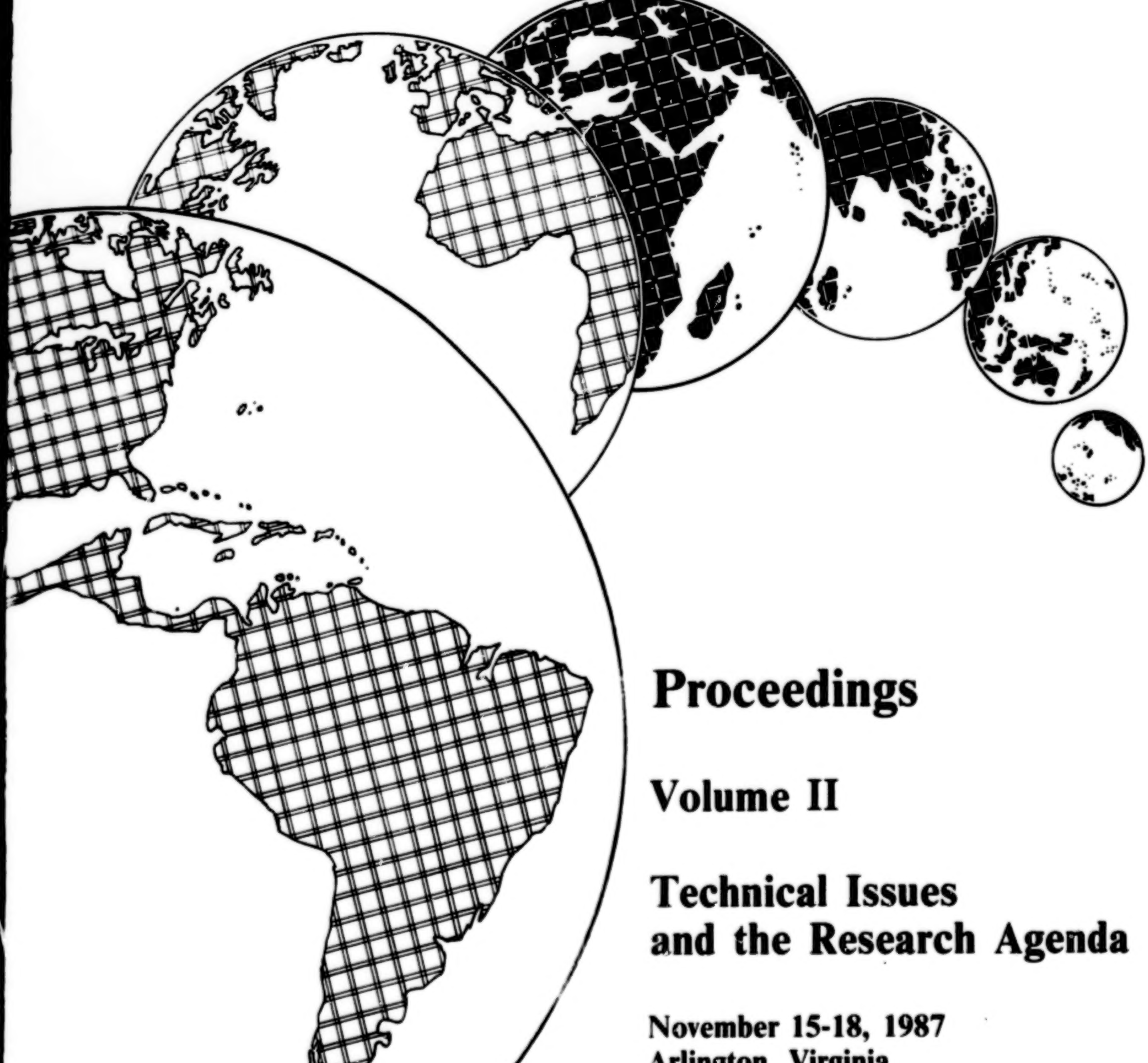
FEB 25 1988

NAS1.2: G29/2/v.2

International Geographic Information Systems (IGIS) Symposium: The Research Agenda



Proceedings

Volume II

Technical Issues and the Research Agenda

November 15-18, 1987
Arlington, Virginia

NASA
National Aeronautics and
Space Administration

COMPLETED

INTERNATIONAL GEOGRAPHIC INFORMATION SYSTEMS (IGIS) SYMPOSIUM

The Research Agenda

November 15-18, 1987

Arlington, Virginia

Proceedings

Volume II

Technical Issues and the Research Agenda

Dr. Robert T. Aangeenbrug, Co-Editor
Yale M. Schiffman, Co-Editor

Convened by the

Association of American Geographers

In Cooperation with
National Aeronuatics and Space Administration
Bureau of Land Management
U.S. Army Engineer Topographic Laboratories
National Oceanic and Atmospheric Admininistration
U.S. Geological Survey
Defense Mapping Agency
Bureau of the Census
Urban and Regional Information Systems Association
American Congress on Surveying and Mapping
World Computer Graphics Association
American Society for Photogrammetry and Remote Sensing
Marine Technology Society
Centre for Earth Resource Management
National Governor's Association
Society of American Foresters
National Agricultural Chemicals Association
Spot Image Corporation
Earth Observation Satellite Company

BLANK PAGE

CONTENTS

Preface xvii
Acknowledgements xix

VOLUME I -- OVERVIEW OF RESEARCH NEEDS AND THE RESEARCH AGENDA

LUNCHEON PRESENTATIONS I-1

GIS Research and Development: A U.K. Perspective. I-3
Lord Roger Chorley, President, Royal Geographic Society, United Kingdom

GIS Development and Support for Earth Sciences. I-8
Dallas Peck, Director, United States Geological Survey

Summary. I-9
Roger Tomlinson, President, Canadian Association of Geographers

PLENARY SESSIONS

SESSION 1. Geographic Information Systems Research Needs: The Federal Agency View I-11

Review of the Federal Research Agenda. I-13
Ronald F. Abler, National Science Foundation

Policy Perspective.

Paul Huray, Office of Science and Technology I-19
Robert Chartrand, Congressional Research Service I-23

Agency Mission Statements

Defense Mapping Agency I-27
Bureau of Land Management I-29

Federal Emergency Management Agency I-30
National Aeronautics and Space Administration I-33

National Oceanic and Atmospheric Administration I-35
U.S. Army Engineer Topographic Laboratories I-38

U.S. Bureau of the Census I-39
U.S. Environmental Protection Agency I-42

U.S. Forest Service I-43
U.S. Geological Survey I-45

SESSION 2. Geographic Information Systems Research Agenda: The Scientific Community Perspective I-47

Waldo R. Tobler, University of California/Santa Barbara I-49
David Rhind, Birkbeck College, U.K. I-53
John Bossler, Ohio State University I-57
Michael Goodchild, University of Western Ontario I-60

SESSION 3. Trends in Hardware and Software for Geographic Information Systems I-67

Introduction. I-69
Michael Dobson, Rand McNally and Company

The Hardware Trends. I-69
Jim Meadlock, Intergraph Corp.

Trends in Geographic Information Systems Software. I-75
Jack Dangermond, Environmental Systems Research Institute

Strategic Planning for Information Systems. I-86
John Lange, IBM Corp.

Comments. I-89
Michael Dobson

Summary. I-90
John Estes, University of California/Santa Barbara

SESSION 4. Spatial Analysis and Artificial Intelligence I-93

John R. Borchert, University of Minnesota I-95
Nicholas R. Chrisman, University of Washington I-101
Michael Dobson, Rand McNally and Co. I-113
Donna J. Peuquet, Pennsylvania State University I-119
Keith Ord, Pennsylvania State University I-128

SESSION 5. Geographic Information Systems Management and Implementation I-139

Introduction. William J. Craig, University of Minnesota I-141

John Scoggins, John Scoggins & Assoc. I-143
John Antenucci, Plangraphics, Inc. I-149
Paul Tessar, Minnesota Planning Agency I-152
Bernard J. Niemann, University of Wisconsin/Madison I-156
William Craig I-162

POLICY I-163

Institutional Research Needs to Maximize Geographic Information Development. I-165
Lisa Warnecke

Litigation Over Information: The Use & Misuse of Maps. I-177
Earl F. Epstein

Geographic Information Systems: Issues Arising from the Proliferation of Information. I-185
Phillip Parent

Urban Geographic Information Systems: Technology Without Solutions. I-197
Daniel M. Parr

GLOBAL SYSTEMS I-207

Connecting Global Science Through Spatial Data and Information Technology. I-209
John J. Kineman and David M. Clark

A World Soil Hydrology File for Global Climate Modelling. I-229
Leonard Zobler

Small-Scale Soil Maps in the United States as Prototypes for Handling Soils in a Global Geographic Information System. I-245
Norman B. Bliss and William U. Reybold

EDUCATION I-257

USMGIS: An Entry Level System. I-259
Donald L. Williams, William R. Partridge, and John R. Burke

Education about and with Geographic Information Systems: Issues and Experiences. I-265
Derek Thompson

GIS Training and the Evolving Marketplace. I-285
Robert V. Maher

GIS in an Undergraduate Geography Programme. I-297
Douglas R. Banting

STANDARDS I-305

GIS Research Needs Identified during the Cartographic Standards Process: Features. I-307
Robert D. Rugg and Warren E. Schmidt

GIS Research Needs Identified During a Cartographic Standards Process: Spatial Data Exchange. I-319
Timothy Nyerges

GIS Research Needs Identified During a Cartographic Standards Process: Cartographic Object Definition and Manipulation. I-331
Harold Moellering

Measuring the Quality of Geographic Databases. I-339
Jeffrey M. Young

List of Attendees I-345

List of Exhibitors I-365

Author Index I-371

VOLUME II -- TECHNICAL ISSUES AND THE RESEARCH AGENDA

CARTOGRAPHIC ANALYSIS II-1

Data Descriptions for Automated Generalization. II-3
M. Kate Beard

Computer-Assisted Map Analysis: Characterizing Proximity and Connectivity. II-11
Joseph K. Berry

Distance Imprecision and Error in Spatial Decision Support Systems. II-23
Marc P. Armstrong

Choropleth Mapping on the Amiga: II-35
Cyrus B. Dawsey, III

Evaluation of the 1:250,000 Digital Elevation Model for Use in a GIS for the Great Smoky Mountains National Park. II-43
James R. Carter

MODELS II-53

The Maturing of GIS and a New Age for Geographic Information Modeling (GIMS). II-55
Jack Dangermond

Towards an Enumeration and Classification of GIS Functions. II-67
Michael F. Goodchild

Application of the Object-Oriented Paradigm to Problems in Geographic Information Systems. II-79
Kenneth J. Dueker and Daniel Kjerne

Issues in the Use of Expert Systems to Manage Uncertainty in Geographic Information Systems. II-89
Vincent B. Robinson, Ross Miller, and Laurie Klesh

Spatial Terms and Spatial Concepts: Geographic, Cognitive, and Linguistic Perspectives. II-101
David M. Mark, Soteria Svorou, and David Zubin

DATA BASE MANAGEMENT II-113

Towards the Automatic Generation of GIS Data Bases from Maps: Research Issues and Status. II-115
Barry J. Glick

Map Algebra: An Object Oriented Implementation. II-127
Kelly K. L. Chan and Denis White

Some Topographic Ingredients of a Geographic Information System. II-151
Richard J. Pike, William Acevedo, and Gail P. Thelin

Updating Multi-Layer Vector Databases. II-165
Jonathan Corson-Rikert

DATA BASE MANAGEMENT THEORY II-175

Data Structures for Quadtree-Addressed Entities on a Spatial Relational Database. II-177
Gregory A. Elmes

Object-Oriented Databases: Database Requirements for GIS. II-189
Max J. Egenhofer and Andrew U. Frank

SPATIAL ANALYSIS II-213

Towards a Spatial Theory. II-215
Andrew U. Frank

Interpolation and Uncertainty in GIS Modeling. II-229
Kenneth C. McGwire and John E. Estes

ARTIFICIAL INTELLIGENCE II-243

Organizational and Knowledge Base Considerations for the Design of Distributed Geographic Information Systems - Lessons from Semantic Modeling. II-245
Vincent B. Robinson and C. Gustav Lundberg

Artificial Intelligence Tools for GIS. II-257
Andrew U. Frank, Douglas L. Hudson, and Vincent B. Robinson

Suitability of Knowledge Representation for GIS. II-273
Piotr Jankowski

Multi-Component Object Search Using Spatial Constraint Propagation. II-281
Sudhakar Menon and Terence R. Smith

GLOBAL POSITION SYSTEMS II-295

Geographical Information Systems and NAD 83. II-297
John D. Bossler

GIS Point Referencing by Satellite and Gravity. II-305
Alfred Leick

REMOTE SENSING II-319

Electronic Browsing for Suitable GIS Data. II-321
Jeffrey L. Star, David M. Stoms, Mark A. Friedl, and John E. Estes

Measurement of Longwave Energy from Selected Urban Surfaces Using Remote Sensing and In Situ Data. II-333
Dale A. Quattrochi

Utility of Topographic Data and Synthetic Aperture Radar Data for Forestry Applications in Mountainous Terrain. II-347
Shih-Tseng Wu

Remote Satellite Imaging and U.S. Information Policy: National Security vs. International Competition. II-357
Mark E. Brender and Manley R. Irwin

STORAGE AND RETRIEVAL II-369

Multi-Colored Quadtrees for GIS: Exploiting Bit-Parallelism for Rapid Boolean Overlay. II-371
Sudhakar Menon, Peng Gao, and Terence R. Smith

Planning for Future Architecture Changes. II-385
Thomas M. Myers

Evolutions on Graphic Workstations and Their Relationship to Micro, Mini, and Mainframe Systems. II-397
Dan Thomas

Space Efficient Hierarchical Structures: Relatively Addressed Compact Quadtrees for GIS. II-405
Peng Gao and Terence Smith

GIS Processing in the National Parks--The UNIX Enhanced PC Environment. II-415
Harvey Fleet and Sharon S. Shin

VOLUME III -- APPLICATIONS AND IMPLEMENTATION

DATA BASE MANAGEMENT:CARTOGRAPHY III-1

Triangulated Date Structures for Map Merging and Other Applications in Geographic Information Systems. III-3
Alan Saalfeld

Toward Geometric Signatures for Geographic Information Systems. III-15
Richard J. Pike

Development of a Digital Data Product Import/Export Capability for a Federal Government GIS. III-27
Gary Krauss

APPLICATIONS III-35

MARKMAP: A GIS for Small Scale Market Area Analysis. III-37
Richard N. Annitto and Robert G. Cromley

DWRIS87 - A U.S. Forest Service GIS Built Around Data Entry by Automated Scanning. III-45
David A. Blakeman

HAZARDS III-57

The Use of USGS 1:100,000 Scale Digital Data as a Base Map Source for Automated Floodplain Management. III-59
Daniel M. Cotter and Richard K. Campbell

A Methodology for Relating Regions of Corrosive Ground Water to Hydrogeologic Variables in the New Jersey Coastal Plain. III-73
J. L. Barringer, R. L. Ulery, and G. R. Kish

Geographic Criteria for the Siting of Low-Level Waste Disposal Sites. III-87
John C. Stewart

Groundwater Pollution Hazard Assessment: A GIS Approach. III-103
James W. Merchant, Donald O. Whittemore, Jerry L. Whistler, Carl D. McElwee, and John J. Woods

HYDROLOGY III-117

The Environmental Protection Agency's Direct/Delayed Response Project: The Role of a Geographic Information System. III-119
William G. Campbell, Gary D. Bishop, M. Robbins Church, Jeffrey J. Lee, Duane A. Lammers, and Leon H. Liegel

Two-Dimensional Display of Geographically-Referenced Three-Dimensional Hydrologic Vector Fields. III-131
T. Barringer, D. Dunn, R. Ulery, and E. Declercq

GIS File Requirements for Regional Scale Water Resources Analysis. III-137
Philip J. Gersmehl and Dwight A. Brown

Map Data Digitizing, Editing and Automatic Hydrological Network Reconstruction. III-149
G. C. Mulaku and G. D. Lodwick

IMPLEMENTATION III-163

A Critical Evaluation of the Proliferation of Automated Mapping Systems in Local Governments. III-165
Michael J. Kevany

Township to State and Vice-Versa. The Potential Minnesota GIS Hierarchy. III-179
Charles G. Parson

The Public Land Survey System: Linking GIS Technology to LIS Applications. III-185
John H. Whiting

The Effective Use of Color in Cartographic Displays. III-189
Hugh W. Calkins and Holly Dickinson

The Quest for a Functional LIS: A Case Study for Prince William County. III-201
Julie A. Phillips

Costs and Benefits of GIS: Problems of Comparison. III-215
Robert F. Gurda, D. David Moyer, Bernard J. Niemann Jr., and Stephen J. Ventura

Implementing GIS Technology for Federal Land Management Tasks - Turning the Dream into the Reality. III-229
Charles A. Grymes

Success Criteria for GIS III-239

David D. Seldon

Critical Management and Technical Considerations When Planning for a GIS Acquisition. III-245
Kenneth M. Green

Determining Your GIS Requirements. The Critical First Step. III-257
Bridget Meckley

Deriving Value from the Modelling and Analysis of Spatial Data. III-271
Giullio Maffini and Wendy Saxton

LAND RESOURCES III-291

The Salt River Project's Land Use Model: A Multi-Dimensional Tool for Allocating Forecasted Land Use Types. III-293
Jacqueline A. Fifield

The Nature of Soil Data in GIS -- Error or Uncertainty. III-307
Peter F. Fisher

Continued Development and Application of a Statewide Land Use Data Base and Analysis System. III-319
James L. Hatchitt, Mark P. Karet, and Stephen Terry

Research Implications of Eleven Natural Resource GIS Applications. III-329
Jerome G. Sullivan and Bernard J. Niemann, Jr.

A Workstation-Based GIS for Land Acquisition and Planning; The Connecticut River Case Study. III-343
Eun Hyung Kim, Michael Waltuch, and Julius Gy. Fabos

Research for Validating Cadastral Data. III-351
Harlan J. Onsrud

Needs of a Large Scale Cadastre at the County Level. III-363
Harry Emrick

DEFENSE III-367

Geographic Information Systems Programming in ADA. III-369
Donald L. Williams, William R. Partridge, and John R. Burke

An Overview of the Army GIS Research Program. III-375
Elizabeth D. Porter

Airland Battlefield Environment (ALBE) Tactical Decision Aid (TDA) Demonstration Program. III-389
Penny R. Capps and Michael J. Tolson

The Development of Small Computer Geographic Analysis Systems for Military Applications. III-407
Barney W. Jones, Michael Rowan, and Nickolas Faust

INTERNATIONAL III-423

The Environmental Generalist and Discipline Experts in LDC Economic Development Planning. III-425
Frederic R. Siegel

GIS Development in Taiwan. III-435
Kang-tsung Chang and Chin-hong Sun

Geographic Information Systems in the Netherlands: Application, Research and Development. III-445
JanJaap Harts and Henk F.L. Ottens

NATURAL RESOURCES III-459

"Under the Rainbow": GIS and Public Land Management Realitites. III-461
Glenn L. Radde

Incorporating GIS Methods into the Natural Resource Manager's Toolbox: Lessons from the Rhode Island GIS Project. III-473
Peter V. August, William R. Wright, and Robert L. Bendick

Sliced Vegetation Maps. III-487
Michael N. DeMers

Computer Mapping for Coal Grade Evaluation. III-499
M. E. Hohn, C. J. Smith, K. C. Ashton, and G. H. McColloch

UNCLASSIFIED PAPERS III-509

Quantitative Dimensions of Neighborhoods for Comparative Research. III-511
Sherman L. Lewis III

System for Inventorying the Land Resource Potential of Tropical Watersheds. III-523
Gustavo A. Antonini, Leonard Zobler, and Roy Ryder

Appropriate Uses of Topology in Geographic Information Mangement Systems. III-533
Ronald D. Exler

Computerized Infrastructure Management in Saint Paul. III-539
Richard A. Person

A GIS for Local Health Services Planning. III-551
Ellen K. Cromley and Robert G. Cromley

BLANK PAGE

PREFACE

These proceedings contain a selection of papers presented at the Association of American Geographers' (AAG) International Geographic Information Systems (IGIS) Symposium which was held at the Hyatt Regency Crystal City Hotel in Arlington, VA, on November 16-18, 1987. The papers cover a wide variety of subject areas related to the conference theme, "The Research Agenda," and describe the current status and future direction of geographic information systems and their application to a number of resource management concerns.

The symposium brought together professionals in such diverse fields as remote sensing, geographic information systems, data base management systems, information systems, urban and regional planning, fish and wildlife management, geography, cartography, artificial intelligence, systems analysis, and resource management. The speakers and audience had an international flavor with representatives from South America, Africa, Europe, Canada, Asia, and the United States. In all, nearly 500 professionals met to exchange information and ideas regarding the role and the research needs and agenda for the development of the next generation of geographic information systems. Fifty exhibit booths displayed the latest geographic information systems hardware, software, and services.

On Monday, November 16, nearly all of the federal agencies involved in the design and development of GIS presented their respective views on agency needs in a plenary session chaired by Ronald F. Abler, National Science Foundation. The policy perspective was provided by Paul Huray, Office of Science and Technology, and Robert Lee Chartrand, Congressional Research Service. The National Aeronautics and Space Administration, the Bureau of Land Management, the National Oceanic and Atmospheric Association, the U.S. Geological Survey, the Defense Mapping Agency, the USETL, the Federal Emergency Management Agency, the Bureau of the Census, the U.S. Department of Agriculture/Forest Service, and the Environmental Protection Agency were all represented. That afternoon, a plenary session chaired by David Cowens, University of South Carolina, explored the GIS research agenda from the perspective of the scientific community.

On Tuesday, November 17, a panel chaired by Michael Dobson, Rand McNally and Company, presented an overview of the trends in hardware and software development of GIS. This plenary session provided the industry view of where GIS seems to be heading. On Wednesday, November 18, a plenary session chaired by Duane Marble, Ohio State University, explored the cutting edge of GIS development in terms of spatial analysis and artificial intelligence. The final plenary session examined some of the practical aspects of GIS management and implementation in a session chaired by William J. Craig, University of Minnesota.

Three keynote addresses were given on Tuesday, November 17, at a well attended luncheon. Roger Chorley, President of the U.K.'s Royal Geographic Society, Dallas Peck, Director of the USGS, and Roger Tomlinson, President of the Canadian Association of Geographers, gave enlightening perspectives on where GIS has been and what is in its future.

The keynote addresses, as well as the plenary sessions, can be found in Volume I.

The meeting closed on a positive note, indicating that there was, indeed, a need to continue dialog by specialists from this wide array of interests. The emphasis should be on improving the communication between the user community and the software, hardware, and analytical support specialists in the field. Thus, this meeting serves as the starting point for a continuing series on the subject.

The interdisciplinary oriented symposium provided a forum for presenting and discussing scientific works in the areas of energy resource management, remote sensing, geographic information systems, other georeferenced data systems, environmental analysis, and applied systems research. Nearly 125 scientists, engineers, planners, and other professionals from around the world contributed to these proceeding.

The proceedings have been organized along subject lines rather than in order of presentation. The editors felt the information would be more useful to the reader if it were organized in this fashion. In Volume I, these papers examine a general overview of the state of the art. Volume II examines the technical issues. In Volume III, the papers explore the topic from an applications focus. Many papers also explore the integration of remotely sensed data with other georeferenced data systems. The proceedings clearly reflect the trends to integrate remotely sensed data with a number of different georeferenced data systems and illustrate an emerging interest among a number of different specialties for use of these integrated data systems in environmental assessment and resource management.

There also appears to be a strong interest in developing countries for the acquisition and utilization of low to moderate cost hardware and software. It is the opinion of the editors that this symposium brought together a mix of professionals--the providers of hardware and software and the users--that needed to communicate with one another. The conference provided the proper setting for an exchange of information which will serve to improve the understanding of all the needs in this emerging field.

June 1988

Dr. Robert T. Aangeenbrug, Co-Editor
Yale M. Schiffman, Co-Editor

ACKNOWLEDGEMENTS

The Association of American Geographers (AAG) would like to acknowledge the contributions of the Science Advisory Committee consisting of Ronald F. Abler (National Science Foundation), Ray Arnold (NASA), Penman R. Gilliam (Defense Mapping Agency), Marcio Barbosa (Instituto de Pesquisas Espacials), John Bossler (Ohio State University), Kurt Brassel (University of Zurich), Robert Lee Chartrand (Congressional Research Service), David Clark (NOAA/NGDC), David Cowen (University of South Carolina), Will Craig (URISA), Doyle Frederick (USGS), Joan Hock (NOAA), Paul Huray (Office of Science and Technology), Duane Marble (Ohio State University), Robert Marx (Bureau of the Census), Jean Claude Mueller (ITC), Bruce Opitz (U.S. Army Engineer Topographic Laboratories), David Rhind (Birbeck College), Ronnie Savegear (ISPRA), Terence Smith (University of California), Duane Sonnenburg (BLM), Lowell Starr (USGS), Waldo Tobler (University of California), and Lofti Zadeh (University of California). The Committee provided useful guidance to AAG's staff and support contractors, Nancy and Yale Schiffman of E.H. Pechan and Associates, Inc.

AAG would also like to acknowledge the Program Advisory Committee for their assistance in developing the sessions. This committee consisted of Ray Allison (USDA), Lorraine Amico (National Governors' Association), Bob Barker (SPOT Image Corporation), Frank Baxter (ACSM) Hugh Bloemer (Ohio University), Walter Boge (ASPRS), Fred Broome (Bureau of the Census), Nicholas Chrisman (University of Washington), Jerry Dobson (Martin Marietta), Ken Dueker (Portland State University), Michelle Ellrod (Marine Technology Society), Earl Epstein (NOAA), Sheila Frye (NOAA), Tom Gilding (National Agricultural Chemicals Association), Barry Glick (Spatial Data Sciences Inc.), Michael Goodchild (University of Western Ontario), Lars Greczy (U.S. Army Engineer Topographic Laboratories), Mason Hewitt (EPA), Robert Jaske (FEMA), Dennis Johnson (U.S. Forest Service), Richard Kott (EOSAT Corporation), Cliff Kottman (DMA), Millington Lockwood (NOAA/NOS), James Merchant (University of Kansas), David Moody (USGS), Richard Mroczynski (EOSAT Company), Denny Parker (Colorado State University), Liz Porter (U.S. Army Engineer Topographic Laboratories), Carl Reed (Delta Systems), Vincent Robinson (University of Calgary), Rolf Schmitt (U.S. Department of Transportation), Duane Sonnenburg (BLM), Theodore Steinke (University of South Carolina), Henry Tom (National Bureau of Standards), William Ubbens (USDA), Tom Usselman (National Research Council), and Fred Wood (Office of Technological Assessment).

AAG thanks those members of both committees who devoted their time and energy to the review and final selection of papers that were presented at the symposium.

AAG also acknowledges the sponsors--the National Aeronautics and Space Administration, the Bureau of Land Management, the U.S.

Army Engineer Topographic Laboratories, the National Oceanic and Atmospheric Administration, and the U.S. Geological Survey--who provided in-kind services, financial support, and contributed to the technical substance of the program.

Finally, special appreciation goes to the National Aeronautics and Space Administration which is making the publication of these proceedings possible.

June 1988

Dr. Robert T. Aangeenbrug, Co-Editor
Yale M. Schiffman, Co-Editor

CARTOGRAPHIC ANALYSIS

BLANK PAGE

DATA DESCRIPTIONS FOR AUTOMATED GENERALIZATION

M. Kate Beard
University of Maine
Surveying Engineering
Orono, ME 04469

ABSTRACT

Manual generalization is a routine cartographic task but its automation presents significant challenges. As a result, automated generalization has become one of the frequently mentioned topics on the GIS research agenda (Boyle 1983; Onsrud and others 1985; Batty 1987). The goal of research in this area must be more than a translation of the traditional manual process. The time and cost of traditional production techniques allows only limited sets of generalizations which appear as different scale maps. Automation of the process should expand our capabilities to allow representation of the data at any resolution or any summary of the data desired by a user. This paper explores automated generalization in a broad context, the problems encountered in automating the process, and research directions to approach the problem.

1. INTRODUCTION

Cartographic generalization has been described as an extremely complex process which requires intuition and special skills acquired through long term training and experience. Capturing this expertise has been one of the challenges of automating the process. Expert systems, which operate by the formalization, structuring and codification of expert knowledge, appear as a natural solution for handling the complexity of automating generalization. This solution, however, encounters a few problems. A review of the abundant literature on generalization reveals that there are no formal and consistent rules for cartographic generalization. One difficulty lies in resolving the confusion and contradictions among the experts. Another difficulty may arise in applying traditional rules for manual generalization to digital representations. The last difficulty stems from the fact that cartographer's actions are less often bound by rules than they are influenced by intuition and circumstance. Each cartographer takes an individual approach to the problem and often each map or individual map objects require different treatment.

This paper presents automated generalization as more than a simple process for minor scale reduction. It must support generalization of attribute as well as spatial information. Ideally, an automated process should be capable of a wide range of scale reductions and modifications to support multiple representations and use of the data. This approach builds on the concept for a single detailed data base (Beard 1987). In this scheme, only the most detailed information is stored and maintained. Smaller scale maps or different resolutions of the data are then produced from this detailed source material as needed. An integrated package of generalization tools can be applied to the detailed data base to create new maps desired by a user. These tools could include processes for selection, aggregation, dimension change, line simplification, and coarsening (area simplification and elimination).

The tools in this case are supported, not by the external knowledge of experts, but by the nature of the data itself. Descriptive measures would be computed from the data and built into the data base. In a sense, this information would become an added dimension or integral part of the data structure such that the data could describe itself to the generalization tools. This information would also assist the user in creating desired products. An advantage in this approach is that peculiarities in the data could be accounted for. Additionally, generalization could be supported as a flexible and dynamic process with not one solution but many, individually tailored by the users.

This paper begins with an investigation of the limitations to effective automated generalization including the confusion and conflicting opinions on the process and differences in traditional and digital map characteristics. This review points to new research directions in the development and organization of data descriptions for a detailed data base. Appropriate digital models for spatial objects and their relations and attributes is a central part of this investigation.

2. CONFLICTS IN EXPERT KNOWLEDGE

The cartographic literature abounds with different approaches to generalization. The diversity, in part, reflects the richness of the English language, yet there are sincere disagreements on what procedures constitute generalization and how these should be performed. Often the same procedures have different meanings in the minds of different cartographers. Much confusion surrounds the process of selection. Robinson and others (1984) stress that selection is not a part of cartographic generalization. They describe it as "the intellectual process of deciding which information will be necessary to carry out the purpose of the map successfully" (p. 150). Other cartographers characterize selection as an integral part of generalization; defining it as the use of objective criteria to reduce the density of objects on the map (Pannekoek 1962; Srnka 1970; Töpfer and Pillewizer 1966; Douglas and Peucker 1973; Lichtner 1979). This commonly held concept of selection coincides for the most part with the process Robinson and others (1984) refer to as simplification: the determination of the important characteristics of the data and the number of individual items to be portrayed on a reduced scale map. The confusion in this instance is twofold. Different procedures are associated with the same term, while different terminology apply to the same procedure. Another confusion occurs in use of the same terms to describe manual and digital procedures. The term smoothing in the manual context implies simplification of cartographic lines. As a digital procedure it involves insertion of points or enhancement of a line (Buttenfield 1984). The lack of agreement among experts on a standard procedural approach to generalization presents a challenge to any formalization of the process.

3. DIFFERENCES IN MANUAL AND AUTOMATED GENERALIZATION

In general, a manual and automated approach share common origins, and purpose. A successful automated process, however, is not likely to be a direct translation of manual techniques. Fundamental differences exist between the representation and manipulation of graphic and digital information and such differences should influence the design of an automated system.

On a traditional map, objects are represented by some set of symbols that occupy physical space. In a digital map, objects are represented mathematically and have no physical presence. Manual generalization is primarily a manipulation of point and line symbols with areas usually not handled as separate objects. Graphically areas are made recognizable by the presence of their boundaries. A person viewing a map makes the connection between bounding lines and a bounded object and perceives an area. Areas are therefore modified by manipulating the lines that bound and define them. The cartographer modifies lines but sees the effect on the areas because of their direct association. If the same bounding lines are transferred to the computer without further structure, the computer only recognizes the set of lines and not the areas. Additional structure is required to establish the connection between lines and areas in a manner similar to the perceptual connection made by the person viewing the map,

In manual generalization symbols are discrete objects with no sub-components. As an example, a line symbol representing a jurisdictional boundary is treated by the cartographer as one object. The line produced from a digital representation can look identical to a line on a traditional map but is actually an assemblage of sub-components. During generalization it is not the line which the viewer sees that is modified, but the line's sub-components or coordinates. The larger entities may enter into the process, but the geometric changes are made at the coordinate level.

In manual generalization, objects are handled within a broad frame of reference. A cartographer can see an entire map sheet and locations of individual symbols with respect to others on the sheet. When pen is set to paper to begin the generalization, the frame of reference narrows. Concentration focuses on the characteristics of individual objects and their relationships to immediately surrounding objects. At the most basic level, the cartographer's attention is focused on a cone of vision a few centimeters around the tip of the moving pen. Small adjustments and conflicts are resolved within this limited scope. At this narrower focus, however, the cartographer still retains an overall comprehension of the larger regional character. Thus the generalization of a cluster of off shore islands at the lowest level may involve decisions to move a line to the right or left of its current position, to remove a small crenulation, exaggerate an island or combine several small islands, but this decision is influenced by the larger context of the general character of the coastline in the mind of the cartographer.

The computer does not easily simulate these larger and variable frames of reference. In an automated system, processing generally focuses on the sub-components of an object without knowledge of the larger entity, its complete form, or its surrounding context. A more global context is possible but local processing is generally promoted to achieve computational efficiency.

4. THE CURRENT USE OF EXPERT SYSTEMS

Several development efforts in automated generalization have looked to the manual process for guidance, but few have captured its subtleties. Expert systems for automated generalization have been offered as one solution, and a number of systems have been proposed and developed (Nickerson and Freeman 1986; Vicars and Robinson 1986). Nickerson and Freeman (1986) proposed a rule based system, MAPEX which consists of rules written in formal grammar and translated into executable FORTRAN 77 subroutines. The rules apply to specific feature types of the USGS Digital Line Graph (DLG) data (boundaries, hydrography, PLSS, transportation). They define the high level decisions of generalization and guide the lower level geometric algorithms which carry out the generalization. Examples of rules include:

- If a lake or pond would have an area less than the minimum polygon area, do not include it, unless there are five or more other small lakes within a radius of ten mm, in which case replace these small lakes with one small lake slightly larger than the minimum polygon area.

- If a road feature is a divided highway, and the new scale is greater than or equal to 100,000, then obtain the neighboring portions of the divided highway and combine them into a single highway feature. (Nickerson and Freeman 1986 p. 551)

These rules formulate generalization tasks in the context of traditional manual operations, and the user has the impression of carrying out generalization in a familiar way. Difficulties arise, however, when differences in digital representations and the computer's limited reference frame are not accounted for. The rules are an extraction of cartographers' expertise but they can lose their effect outside the context of a particular geographic situation and without the cartographers' frame of reference. Displacement is a common generalization function handled quite easily by the cartographer. Translated as an automated routine, it can create chain reactions of collisions and relocation of objects. As Nickerson and Freeman point out, under certain conditions, extensive relocations are required. In manual generalization, the cartographer has the information and larger frame of reference to avoid such trial and error maneuverings. The rules may replicate the cartographers actions, but crucial information is missing.

The rules have other problems. Even limited to the four major categories of the DLG data, the development of rules to handle all possible situations and relations can be overwhelming. It is difficult to design rules that anticipate or respond effectively to every situation. In considering the first example above, there could be a case in which four small lakes are within 9.5 mm of each other and the rule would not apply although it may be appropriate. An alternative is to base a generalization action on the particular characteristics of the data at any location rather than on external rules.

5. DATA DESCRIPTIONS TO SUPPORT GENERALIZATION DECISIONS

The majority of decisions in manual generalization are matters of judgement made on a case by case basis. Cartographers follow some general rules, but can adjust these at any time to handle unique situations and characteristics of the data. It is these unique characteristics and spatial relations of the data that in essence direct the cartographer's actions. The logical parallel is that these characteristics of the data should direct the automated process as well, but this potential symmetry is not easily achieved. The visual assimilations and comparisons which humans appear to perform quite simply are, as yet, excruciatingly difficult to replicate in the computer. The cartographer very quickly processes multiple characteristics of individual objects such as their size, shape, and attribute values along with their spatial relationship with other objects. As the desired degree of generalization changes, the cartographer can easily assimilate different sets of characteristics and relations which may apply. The research challenge lies in identifying characteristics of the data that are pertinent to its generalization and translating these into a form which is easily digested by the computer and can be effectively utilized by automated generalization tools. Investigation of these characteristics could begin with examination of the information a cartographer gleans from a map and uses to make decisions.

5.1 Previous Research

Some research has already been carried out along these lines. Marino (1979) provided empirical evidence of the information both cartographers and non-cartographers use to retain the essential character of lines as they are generalized. Participants in her study identified points along the line referred to as critical points. The majority of these points were points of tangency, points of intersection, and points of inflection. The Douglas-Peucker (1973) line simplification algorithm uses points of maximum deviation from the local trend of the line as the geometric characteristic for directing the process. Buttenfield (1987) expands on these previous definitions with digital definitions which preserve the scale dependent geometry of the data. These definitions were developed by isolating and measuring features within a line at all levels of resolvable detail. These digital definitions can be used to direct and adjust a line simplification process as the character of a line changes.

The fractal dimension D (Mandelbrot 1982) of a curve provides another measure of line character. For a curve, D can take on values between 1 and 2. Values of D closer to 1 can indicate the presence of long range variation while values closer to 2 are generally associated with short range variation. Short range variations are most sensitive to generalization and scale reduction, so high D values may indicate the potential for significant change in the data with minor generalization. Changes in the fractal dimension from one region to another can also signal the need for different generalization decisions. The fractal dimension is thus another measure which provides information about the character of a line that could be utilized by a line simplification tool. The fractal dimension also provides an index of surface variation and thus information for the generalization of surfaces.

Different characteristics of the data influence decisions as the degree of generalization increases. Critical points can supply the necessary information for minor generalization, but they do not provide sufficient information for radical scale reduction. For large scale reductions, critical points may no longer be critical since an entire feature containing the points may become insignificant. The size, shape and interaction of objects take on greater importance. In other words, we switch from the consideration of points along a line to consideration of components of areal units and the units themselves. Perkal (1966) provided an operational definition for identifying these components. He identified the essential or critical components of an object as those whose width exceeded a distance epsilon (diameter of a circle) and essential objects as those large enough to inscribe a circle of diameter epsilon. This criteria provides information for the simplification and elimination of two dimensional objects.

5.2 Missing Data Descriptions

Size, shape, orientation and other measures of isolated objects are useful descriptions of the data, but measures of interaction and density of map objects are also essential. The interaction of the size and spacing of objects influences the cartographer's decision on their treatment. If objects are very densely spaced, they cannot all be legibly shown at small scales. Some objects must be removed or they may be grouped to form larger spatial units. Given the spatial arrangement of objects in Figure 1, the cartographer might decide to a). eliminate all objects, b). aggregate them into one mass or c). merge them into two clusters during generalization.

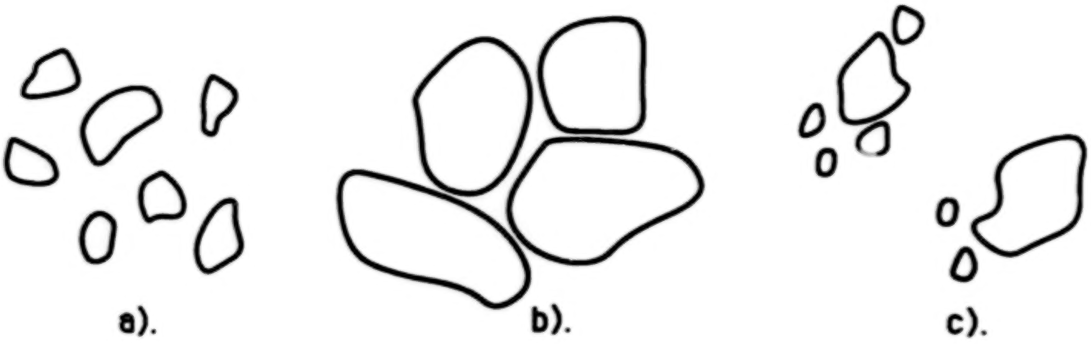


Figure 1 Different arrangements in the size and spacings of objects which would influence their generalization.

The intricate relationships of size and spatial proximity which enter into these decisions can be the most difficult to replicate or model in the computer. Summary measures of spatial proximity and density for a coverage are not adequate as they can obscure the local densities and spatial patterns most pertinent to generalization decisions. Data descriptions or structures are needed that can capture and characterize a local spatial context.

5.3 Attribute and Topological Descriptions

The information cartographers use in making generalization decisions is not limited to geometric descriptions. Topological and attribute characteristics of the data enter into these decisions as well. Cartographers make different decisions depending on whether objects belong to the same or different classes or on the importance of individual objects for a specific map purpose. There may be some natural attribute hierarchies in the data which would influence generalization, but different summaries of attribute characteristics desired by a user should also be accomodated. As an example a local road might logically be aggregated into a higher class of transportation facilities, but it might also be part of the class of all impermeable surfaces in an aggregation of the data into permeable and non-permeable areas.

Topological measures such as adjacency, connectedness, and number of contacts supply equally important information for generalization. In many cases, it is the unique interaction of metric, topological and attribute information which generates a particular generalization result. As an example, the combined size of an object, its attributes, and the degree to which it is surrounded by another object could dictate its fate in the generalization process. Weibel (1987) has used the combined geometry, topology and attributes of structure lines to form a structure line model (SLM) for the generalization of landforms.

6 CONCLUSION

Research in the development of data descriptions to support automated generalization shows progress. Descriptions of line character have received the most attention, but additional descriptions of the data and data interaction are needed. In particular, research should investigate methods to describe the multiple interactions between spatial objects. Identification of appropriate descriptions of the data, however, is only part of the task. If automated generalization is to approach a real time process, a large collection of unstructured descriptions that the computer must check in order to proceed is no solution. Research in this area should focus on the formal structuring of spatial, attribute and topological information in such a way that it can efficiently direct the automated generalization process. Research along these lines is not limited to the support of automated generalization but applies generally to the improvement of GIS functions.

REFERENCES

Batty, M. 1987. 'Research Priorities in the Development of Geographic Information Systems,' Proceedings NSF/NSERC Conference, Aberdeen.

Beard, M.K., 1987. 'How to Survive on a Single Detailed Data Base,' **Proceedings, AUTO-CARTO 8.** 211-220.

Boyle, A. R. 1983. **Final Report of a Conference on the Review and Synthesis of Problems and Directions for Large Scale Geographic Information System Development.** NASA Contract NAS2-11346.

Buttenfield, B.P. 1984. Line Structure in Graphic and Geographic Space. Unpublished PhD Dissertation, University of Washington.

Buttenfield, B.P 1987. 'Automating the Identification of Cartographic Lines,' **American Cartographer**. 14: 7-20.

Douglas, D.H. and Peucker, T.K. 1973. 'Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature,' **Canadian Cartographer**. 10 112-123.

Lichtner, W., 1979. 'Computer Assisted Processing of Cartographic Generalization in Topographic Maps,' **Geo-Processing**, 1:183-199.

Mandelbrot, B. 1982. **The Fractal Geometry of Nature**. New York:Freeman.

Marino, J.S., 1979. 'Characteristic Points and their Significance in Cartographic Line Generalization; and Empirical Study,' **The Canadian Cartographer**. 16: 70-80.

Nickerson, B. G., and Freeman, H. 1986. 'Development of a Rule-Based System for Automatic Map Generalization,' **Proceedings, 2nd International Symposium on Spatial Data Handling**. 537-556.

Onsrud, H. J., Clapp, J and McLaughlin, J. 1985. **A Report of the Workshop on**

Fundamental Research Needs in Surveying, Mapping, and Land Information Systems. Virginia Polytechnic Institute, Blacksburg, Virginia.

Pannekoek, A.J., 1962. 'Generalization of Coastlines and Contours,' **International Yearbook of Cartography.** 2: 55-75

Perkal, J. 1966. "An Attempt at Objective Generalization,' in **Discussion Paper 10.** Ann Arbor, MI: Michigan Inter-University Community of Mathematical Geographers. translated by W. Jackowski.

Robinson, A.H., Sale, R., Morrison, J.L., and Muercke, P. 1984. **Elements of Cartography.** 5th edition. New York, N.Y.: John Wiley and Sons.

Srnka, E. 1970. 'The Analytical Solution of Regular Generalization in Cartography,' **International Yearbook of Cartography.** 10: 48-62.

Töpfer, F., and Pillewizer, W., 1966. 'The Principles of Selection,' **Cartographic Journal.** 3: 10-16.

Vicars, D.W., and Robinson, G.J. 1986. 'Generalization from Large to Medium and Small Scale Ordinance Survey Maps Using Expert Systems Techniques,' **Proceedings, AUTO-CARTO London.** 267-275.

Weibel, Robert. 1987. 'An Adaptive Model for Automated Relief Generalization', **Proceedings AUTO-CARTO 8** 42-49.

COMPUTER-ASSISTED MAP ANALYSIS: CHARACTERIZING PROXIMITY AND CONNECTIVITY

Joseph K. Berry

Spatial Information Analysis, Incorporated

Science Park, New Haven, Connecticut, USA

ABSTRACT

Cartographic models addressing a wide variety of applications are composed of fundamental map processing operations. The major classes of operations involve reclassifying map categories, overlaying maps, characterizing neighborhoods and determining distance and connectivity. This paper presents the basic concepts and considerations involved in computer-assisted analysis of distance and connectivity. The concept of simple distance is first expanded to one of proximity, and then to one of movement through geographic space considering absolute and relative barriers. Connectivity among map features is first discussed as straight lines, then expanded to consider serpentining interconnections and connection density.

1. INTRODUCTION

Geographic Information Systems (GIS) include processing capabilities relating to encoding, storage, analysis and display of mapped data. The analytical part of many of these systems are organized as a set of primitive operations forming a generalized cartographic structure (1, 2, 3). In this context, the operations are similar to those of traditional mathematics, such as addition and subtraction, which are logically sequenced on variables to solve specific application equations. In map analysis, however, the unknowns are represented as entire maps, defined by many numbers, and stored as either a set line segments or grid cells.

To provide a flexible processing structure, each operation accepts input and generates output in the same format. This is accomplished by requiring each analytical procedure involve:

- o retrival of one or more maps from the data file;
- o maniplulation of that data;
- o creation of a new map whose values are a function of the values on the input map(s), yet have the same geographic registration; and,

- o storage of the new map for subsequent processing.

The cyclical nature of this processing structure is similar to the "evaluation of nested parentheticals" in traditional algebra. The logical sequencing of a series these primitives on a set of maps forms a cartographic model of a specified application, evaluated for a particular area. As with traditional algebra, fundamental techniques involving several primitive operations can be identifed. For example, the equation

$$\text{percent change} = \frac{(\text{new price} - \text{old price})}{(\text{old price})} * 100$$

could be used to calculate the change in the price of a shirt. Or the same equation could be used to calculate the percent change in the value of a parcel of land. In a similar manner, a map of percent change in land values for an entire area can be expressed as

```
COMPUTE NEWVALUE.MAP MINUS OLDVALUE.MAP
TIMES 100 DIVIDEDBY OLDVALUE.MAP
FOR PERCENTCHANGE.MAP
```

where the variables "newvalue.map" and "oldvalue.map" are digital maps.

Within this model, data for current and past land values are collected and digitized. These data are evaluated, as shown above, to form a solution map of percent change. The simple model might be extended to provide coincidence statistics, such as

```
CROSSTABULATE ZONING.MAP WITH PERCENTCHANGE.MAP
```

for a table summarizing the spatial relationship between zoning and the change in land value (that is, which zones have experienced the greatest decline in market value; or, which the greatest increase?). The model might be further expanded to include geographic searches, such as

```
RENUMBER PERCENTCHANGE.MAP
ASSIGNING 0 TO -20 THRU 20
FOR BIGINCREASES.MAP
```

for a map isolating those areas experiencing large changes in land value (exceeding ± 20 percent).

Within this processing structure, each primitive operation may be regarded as an independent tool. Depending on the nature of the processing and the results obtained, four major classes of map analysis operations may be identified (4). These include:

- o reclassifying map categories by simply reassigning the values of an existing map;
- o overlaying maps by assigning a new value to every location that is the function of the values associated with that location on two or more maps;

BEST COPY AVAILABLE

- o characterizing cartographic neighborhoods as a summary of the values occuring within a specified distance about a location; and,
- o determining distance and connectivity among locations based on the juxta-positioning of map features.

The basic concepts and considerations in the last class of operations serve as the focus of this paper. All of the analyses demonstrated were generated using the Professional Map Analysis Package (pMAP) (5) by Spatial Information Systems, Incorporated. The line graphics displays were created using the SURFER system (6) by Golden Software, Incorporated. The examples presented were initially developed by the consulting firm of Spatial Information Analysis, Incorporated (7). All of the processing was done using a standard IBM PC-AT personal computer.

2. DETERMINING DISTANCE

Most geographic information systems contain analytic capabilities for reclassifying and overlaying maps. These operations address the majority of applications that parallel manual map analysis technique (8). However, with the advent of modern computers and the need to more fully integrate spatial considerations in analysis, new techniques which quantitatively assess neighborhood characteristics and measure distance are emerging. These new procedures provide analytic capabilities beyond those of manual processing. The concept of distance, for example, has historically been associated with the "shortest line between two points." While this measure is both easily conceptualized and implemented with a ruler, it is frequently insufficient in a decision-making environment. A straight line may indicate the distance as-the-crow-flies, but offer little information for the walking or hitchhiking crow who encounters a complex pattern of spatial barriers. Equally important to most travelers is to have the measurement expressed in more relevant terms, such as time or cost, rather than geographic units, such as feet or miles.

The basis of any distance measuring system requires two components- a standard measurement unit and measurement procedure. In manual measurement, the "tic-marks" along the ruler define the measurement unit. The measurement procedure involves laying the ruler down along a straight line connecting two points and counting the number of marks separating the points. The scale of the map is then used to convert the ruler measurement into geographic units. Consideration of another pair of points repeats the process. In a computerized geographic information system, this procedure is severely limited. In effect, the ruler represents a row along an imaginary reference grid composed of othogonal lines defined by the ruler's edge and tic-marks. The physical repositioning of the

BEST COPY
AVAILABLE

ruler has the effect of rotating the entire reference grid. In a digital map this would require the recomputation of the coordinates of every point in the map, an ornerous task at best. An alternative approach fixes the reference grid and uses the Pathagorean Theorem to compute the distance, based upon the displacement between the points along the two axes of the grid. Though this procedure is readily implemented in computer environments, it cannot easily handle distances between features composed of several points, nor the "real world" consideration of the effect of barriers to movement.

The standard measurement unit used in most computer systems is the "grid space" implied by superimposing an imaginary uniform grid over a geographic study area. The distance from any location to another is computed as the number of interveening grid spaces. A frequently employed measurement procedure involves expanding the concept of distance to one of proximity. Rather than sequentially computing the distance between pairs locations, concentric rings are established around a location, or set of locations. This is analogous to the wave pattern generated when a rock is thrown into a still pond, with each wave representing an additional one grid space step. The resuling map indicates the shortest straight-line distance to the "target" location for all other locations in the mapped area (i.e., simple proximity). A more complex map could be generated by considering several target locations- in effect, throwing a handfull of rocks into a pond. The result would be a map of proximity indicating the shortest straight-line distance to the nearest of the several target locations.

In many applications, however, the shortest route between locations may not always be a straight line. And even if it is straight, the geographic length of that line may not always reflect a meaningful measure of distance. Rather, distance in many applications is best defined in terms of movement expressed as travel-time, cost or energy, which may be consumed at rates that vary over time and space. Distance-modifying effects may be expressed cartographically as "barriers" located within the space in which the distance is measured. Note that this implies that distance is the result of some sort of movement over that space and through those barriers. Two types of barriers can be identified as to how they affect movement. Absolute barriers are those which completely restrict movement and, therefore, imply an infinite distance between the points they separate, unless a path around the barrier is available. For example, a river might be regarded as an absolute barrier to a non-swimmer. To a swimmer or boater, however, that same river might be regarded a relative, rather than absolute barrier. Relative barriers are passable, but only at a cost which may be equated with an increase in physical distance (e.g. longer time).

BEST COPY AVAILABLE

Absolute and relative barriers may be expressed as map values indicating their ability to impede movement (i.e. friction coefficients). For example, movement by automobile is effectively constrained to a network of roads with various speedlimits. Effective proximity is measured along those roads at the respective speeds resulting in a map of travel-time. From a less conventional perspective, proximity can be expressed in such terms as accumulated cost of powerline construction from an existing trunkline to all other locations in a study area. The resulting map can be a function of a variety of social and engineering factors, such as adverse terrain and visual exposure, expressed as absolute and relative barriers.

The ability to move, whether physically or abstractly, may vary as a function of the movement itself. The direction of travel, an attribute of the movement itself, can change the effect of a barrier. For example, a steep slope may significantly slow a hiker moving up that incline, but moving across the slope is similar to movement along the flat. Another possible modifying factor is accumulation. After hiking a long distance, "molehills" tend to become nearly insurmountable mountains. A third aspect of movement which may affect the ability of a barrier to restrict movement is speed. Suppose a short hill is encountered by a heavily laddened truck. If it is approached at a high rate of speed, it is quickly ascended. However, a slow approach will result in a seemingly endless time the ascent. This condition almost always occurs when you are behind the truck and in a hurry.

3. CONNECTIVITY

A distance-related group of operations involve the delineation of paths, or connectivity, among locations. In the case of simple distance, these paths locate the shortest straight line considering only planar directions. Another technique traces the steepest downhill path along a three-dimensional surface. For example, the steepest downhill path along a topographic surface will indicate the route of water runoff. For a surface represented by a travel-time map this technique traces the optimal, or quickest, route. If an accumulation cost surface is considered, the minimum cost route will be located. In each instance, the optimal path identifies the actual route of the wavefront defining effective proximity (as described in the previous section) that responds to various relative and absolute barriers.

This procedure can be expanded to characterize a network of interconnections by assigning a value indicating the number of optimal

BEST COPY AVAILABLE

paths traversing each location in a study area. For example, if the steepest downhill paths of all locations on a terrain surface are considered, a "density surface" of water flow is created. This map identifies locations of water channeling (i.e. gullies leading to streams) and minimal water flow (i.e. ridge tops). An extension of this technique weighs each optimal path according to its importance. In this instance, the result not only responses to the number of connections, but the nature of the set of connections. For example, maps of crop and soil types could be used to determine the phosphate loading potential from each location in a study area and this information used to weight each optimal path determining the "weighted" optimal path density surface.

4. EXAMPLE

To illustrate how primitive map analysis operations can be combined to perform complex analyses, consider the following cartographic model. Simple distance, proximity, movement and connectivity is characterized for the ranch location in the lower-left portion of the study area. The right side of Figure 1 depicts the shortest straight-line distance from the ranch to everywhere in the study area (darker tones indicating locations farther away). Note the concentric rings (shown in the accompanying planimetric plot) which form the characteristic wave pattern emanating from the ranch. This map was created by entering (in pMAP command language)

```
SPREAD RANCH TO 30 FOR PROXIMITY1
```

causing the computer to retrieve a map of the ranch's location from storage, compute the successive rings to 30 grid spaces and store the result as a map called proximity1. The left side of the Figure

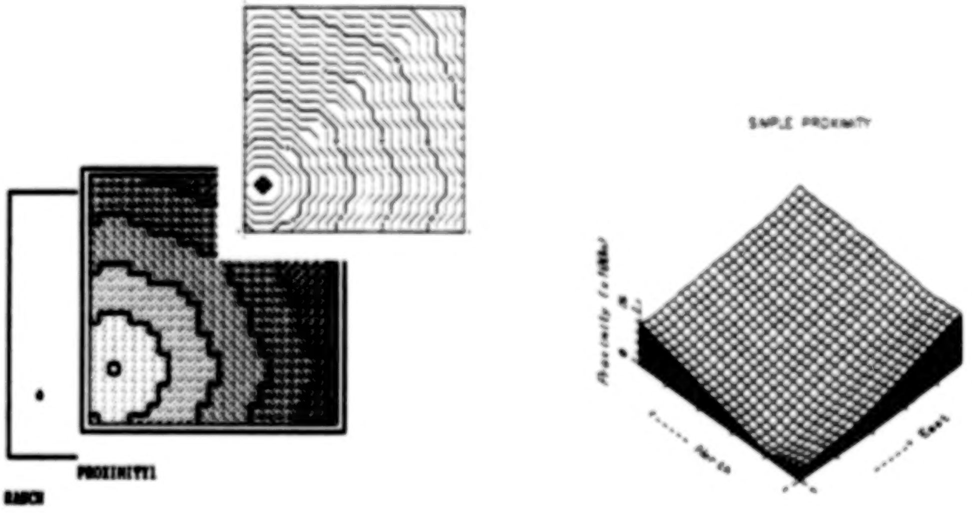


Figure 1. Characterizing simple proximity.

BEST COPY AVAILABLE

shows a 3-dimensional plot of the same data. This bowl-like configuration is characteristic of a simple proximity surface. It indicates the accumulation of distance, increasing at a constant rate of one grid space throughout the study area. The furthest point is 3.0 kilometers away (30 of the one hundred meters in length grid spaces), located in the northeast (upper-right) corner of the study area.

If more than one starting location were considered, there would be several basins. Each basin would be surrounded by increasing distances until a ridge is formed between each target location. A ridge defines the locations that are equidistant to two origins (i.e., the perpendicular bisector). Locations on either side of the network of ridges are closer to one of the origins. For example, if all of the houses in the study area were considered, instead of just the ranch, a proximity to housing map would be generated with each cell indicating how far away it is from the nearest house. Similarly, a map of proximity to water could be generated by "spreading" from a target map indicating wetlands, lakes, and streams. This would result in a complex accumulation surface, similar to a togographic surface, with the water features forming "valley bottoms" (zero grid spaces away from water) and the surrounding "hills" indicating increasing distance away. The ridges would indicate equidistant locations. However, in this instance the ridges would rarely form extended straight lines as they conform to the shape of each water feature.

Figure 2 characterizes weighted proximity considering water

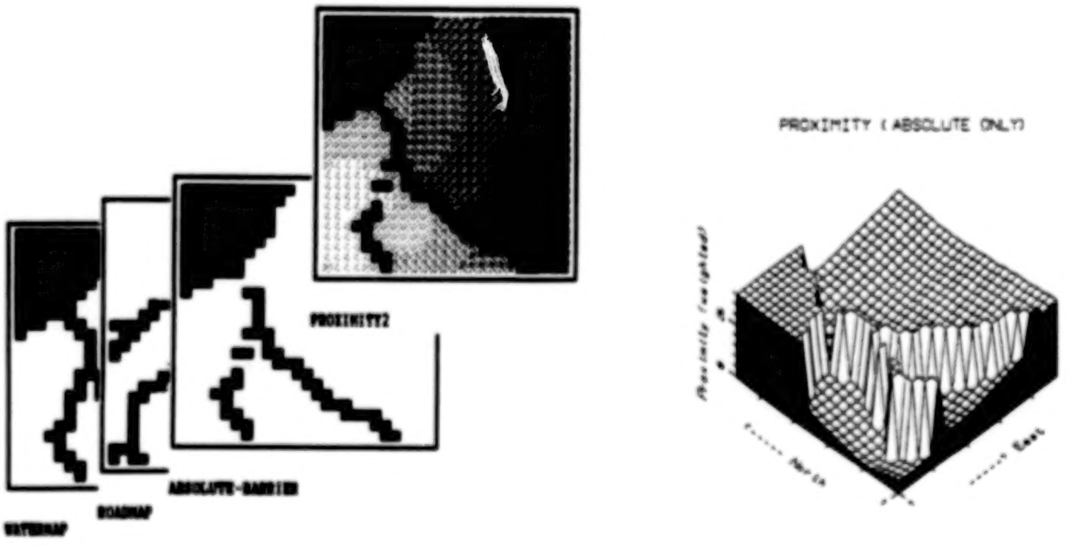


Figure 2. Proximity considering absolute barriers.

BEST COPY AVAILABLE

features as absolute barriers to a hiker-- mom wouldn't want you to get your shoes wet. The right side identifies these barriers asmodified to consider bridges, from a map of roads, as places where you can actually walk on water. Note the important breaks in the otherwise continuous stream barrier in the center portion of the map. The left side of the figure contains a plot of the accumulation surface created by entering

SPREAD RANCH TO 40 OVER ABSOLUTE FOR PROXIMITY2

The barriers are shown as the farthest away locations (actually infinitly far away unless you are willing risk mom's rath and come home with wet shoes). This surface is similar to the one shown in Figure 1 with the same constant rate of accumulation, however, areas on the other side of a barrier are reached through a longer route which circumvents the barriers. The farthest away acessible location is 3.7 kilometers (37 grid spaces) located in the southeast (lower-right) corner.

The effects of both absolute and relative barriers is shown in Figure 3. The map of relative barriers (inset(a)) identifies the ease of hiking throughout the study area. It was assumed that it takes one minute to walk 100 meters (one grid space) in open terrain, and twice that long to go the same distance in the forested areas (you must constantly be alert for lions and tigers and bears). An adjustment was made for roaded areas within the forest, making them as easy to cross as open terrain. Inset (b) contains a plot of the accumulation surface created by entering

SPREAD RANCH TO 60 OVER ABSOLUTE THRU FRICTION /
FOR PROXIMITY3

This surface is much more complex than the ones in Figures 1 and 2,

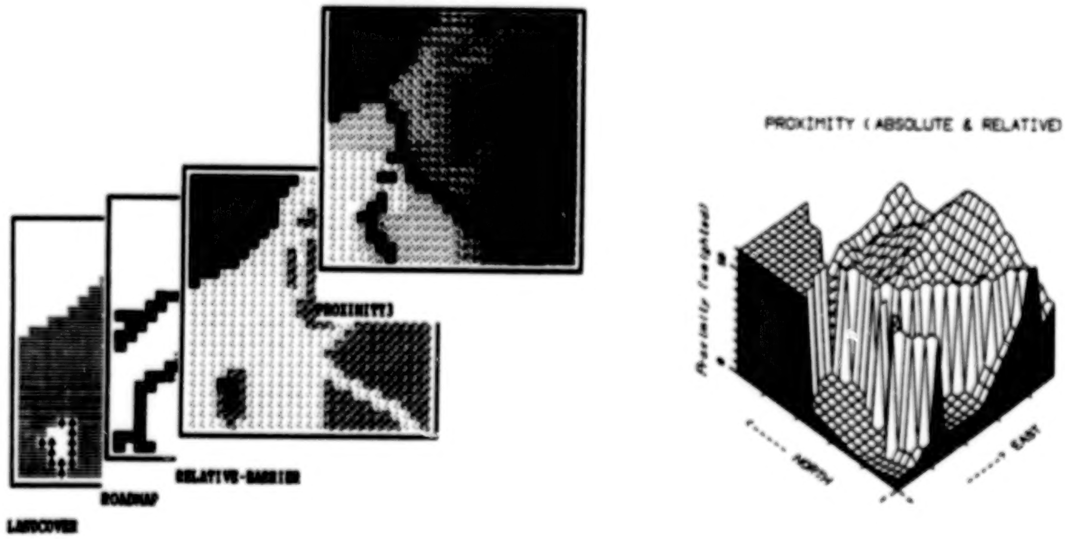


Figure 3. Weighted proximity considering both absolute and relative barriers.

BEST COPY AVAILABLE

as distance measurement responds to the unique pattern of absolute and relative barriers. This has the effect of using a "rubber ruler" where it is streached for locations of easy hiking and compressed for the forested areas. The units of measurement are not fixed geographic steps, as in the previous examples, but units of time per grid space which vary in different terrain. The measurement procedure involving a progressive wavefront actually considers all possible paths from the ranch to all locations, under all conditions, and retains the accumulated time of the shortest one. The farthest accessible location is 57 minutes, located in the extreme center-right (east) portion of the study area.

Each of the three proximity maps identify the shortest distance from the ranch to all other locations. The first one assumes a straight-line connection. The second one relaxes the requirement that all connections must be straight, but assumes that all movement occurs at the same rate. The last one relaxes both requirements. However, none of these maps actually indicate the route, or connection path, defining the shortest distance. The right side of Figure 4 identifies the "best" path from the ranch to a cabin in the northeast (upper right) portion of the study area. This route retraces the movement of the distance measuring wavefront which generated the travel-time surface in Figure 3. It was created by entering

STREAM CABIN OVER PROXIMITY3 FOR PATH

Once an accumulation surface is created, the optimal path from any location to the target(s) can be identified in a similar manner.

The left side of Figure 4 depicts the simultaneous consideration of

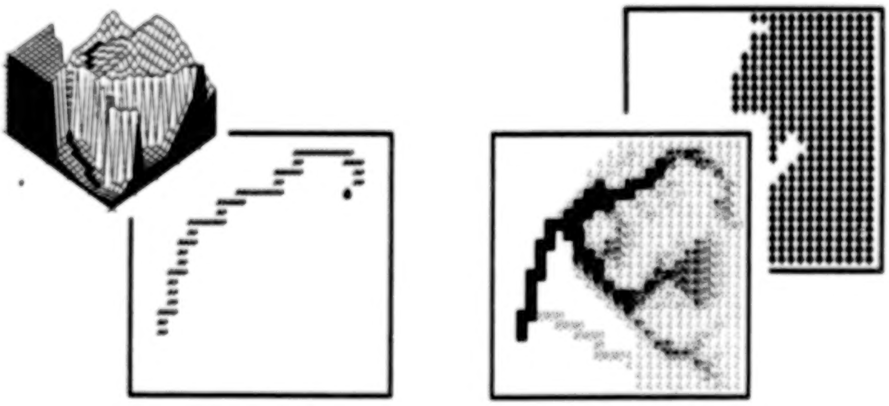

PATH
PATH-DENSITY

Figure 4. Establishing optimal paths.

BEST COPY AVAILABLE

several "streamers." It was created by entering

DRAIN FORESTS OVER PROXIMITY3 FOR PATH-DENSITY

In this instance, all of the paths from the forested areas to the ranch were determined and the number of paths passing through each cell tabulated to form an "optimal path density surface." Note the three distinct trunk-lines servicing the extensive forested area in the northeast portion of the study area. The outer two paths reflect the relative ease of travel along the forest roads in these areas. The middle tunk-line clearly indicates that the remote forest locations are best accessed from the lower road.

4. SUMMARY

Computer-assisted map analysis is providing new techniques for processing mapped data. This paper introduced a cartographic modeling structure composed of fundamental analytic operations analogous to traditional mathematics. Within this structure primitive operations are logically sequenced on map variables similar to the algebraic solution of equations for unknowns. One class of operations, measuring distance and connectivity, was discussed in detail. This advanced set of capabilities extends the concept of distance beyond our usual definition of "shortest straight line." A more useful concept of proximity, identifying the distance from a location(s) to everywhere, was presented. The concept was expanded to one of movement through geographic space considering interveening absolute and relative barriers. This more realistic concept of distance relaxes the assumption that all movement is in straight lines, yet retains the assumption that we are most interested in the shortest route. Connectivity operations identify this optimal route defining the shortest, but not necessarily straight connection among features. The concept was expanded to one that simultaneously considers the interconnections among several features, resulting in optimal path density surfaces. Examples of these concepts using a simplified data base were presented.

References

1. BERRY, J.K., 1987: Computer-Assisted Map Analysis: Potential and Pitfalls, Photogrammetric Engineering and Remote Sensing, Vol. 53, No. 10, October 1987, pp. 1405-1410.
BERRY, J.K., 1986: Learning Computer-Assisted Map Analysis, Journal of Forestry, October issue, pp. 39-43.

2. DANGERMOND, J., 1986: The Software Toolbox Approach to Meeting User's Needs for GIS Analysis, Proceeding of the Geographic Information Workshop, Atlanta, Georgia, American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, pp. 66-75.

BEST COPY AVAILABLE

3. TOMLIN, C.D., 1983: Digital Cartographic Modeling Techniques in Environmental Planning, doctoral dissertation, Yale University, School of Forestry and Environmental Studies, New Haven, Connecticut.
4. BERRY, J.K., 1987: Fundamental Operations in Computer-Assisted Map Analysis, International Journal of Geographical Information Systems, Vol. 1, No. 2, pp. 119-136.

5. pMAP, 1986: The Professional Map Analysis Package (pMAP) User's Manual and Technical Reference, Spatial Information Systems, Inc., 9088 Tiffany Park Court, Springfield, Virginia.

6. SURFER, 1987: SURFER Informational Manual, Golden Software, Inc., P.O. Box 281, Golden, Colorado.

7. SIA, Spatial Information Analysis, Inc., 5 Science Park, New Haven, Connecticut.

8. McHARG, I.L., 1969: Design with Nature, Doubleday/Natural History Press, Doubleday and Company, Garden City, New York.

BLANK PAGE

Distance Imprecision and Error in Spatial Decision Support Systems

Marc P. Armstrong
Departments of Geography and Computer Science
316 Jessup Hall
University of Iowa
Iowa City, IA 52242
BLAMMGPD@UIAMVS.BITNET

ABSTRACT

Locational models are being used with increasing frequency in geographic information systems. Many such models use distance measurements in calculating solutions. A number of commonly used distance measurement procedures are used to examine whether results are influenced by the various estimates. In one set of analyses, the objective function increases with the scale factor of the source map. Additionally, the geometrical configurations yielded by the analyses vary among the distance estimation procedures.

1.0 INTRODUCTION

Geographic information systems (GIS) use sophisticated data capture and display functions to provide assistance in solving a range of social and environmental problems. At the same time, locational analysis methods continually improve, and system designers and academic researchers are now integrating these analysis procedures into GIS toolboxes (Densham and Armstrong, 1987; Elmes and Harris, 1986; Lupien, Moreland, and Dangermond, 1987). This melding of analytical GIS and locational modeling capabilities has led to the formulation of spatial decision support systems (DSS), in which alternative locational decisions can be explored and evaluated (Armstrong, Densham, and Rushton, 1986; Davis and Grant, 1987).

As GIS and DSS technology is brought to bear on important, and potentially threatening, environmental and social issues, the results and recommendations stemming from the application of the technology will be carefully examined. Obtaining insight into the nature of imprecision and error is important in this context, because if a DSS is to be used effectively, the effects of error and imprecision on results must be recognized.

This paper examines one facet of error. The specific goal is to evaluate the role of commonly used distance estimates in calculating the objective functions and resulting geometrical configurations of p-median analysis procedures used to support spatial decision-making.

2.0 BACKGROUND AND PROBLEM ELABORATION

2.1 Concepts of Distance

Distance has long been cited as a fundamental concern of quantitative geography (Nystuen, 1963). Many researchers employ distances (e.g. miles) based on the general Minkowski metric (Kuiper, 1986). Others have used alternative distance measures such as time-distance, and cost-distance, both of which are functional measures of effort (Falk and Abler, 1980). While these functional distances are important, metric distances continue to be used because they are convenient to use and because other measures of distance often can be derived from them.

2.2 Spatial Data Organization and Distance Computation

Although geographic information systems store data in various ways, many define objects using a vector approach (Peuquet, 1984; White, 1984). When data are vector coded, objects such as road networks are often defined by chains of points with associated topological linkages. Network distances are computed in vector databases by summing scaled interpoint distances within each chain. Given measurement costs, analysts may wish to rely on existing data to compute distances. For example, the U.S. Census Bureau has developed digital data products to support its operations (e.g. DIME), and the USGS is actively involved in encoding and disseminating digital data. These and other digital sources will play increasingly prominent roles in spatial modeling during the coming decade.

2.3 Error in Digital Representations of Lines

Peucker (1975) has outlined a general theory of cartographic lines. The theory is especially pertinent to the treatment of line complexity. Chrisman (1982) has developed a preliminary theory for assessing cartographic error and its measurement. Other researchers (e.g. Muller, 1987) have addressed specific dimensions of error and have identified a series of error sources. These sources can be placed into the following categories:

Sampling density. The well-known (e.g. Perkal, 1966; Buttenfield, 1985) Steinhaus paradox shows that as smaller sampling intervals are used to measure a sinuous feature, it apparently increases in length. Sinuosity which was missed in previous intervals is now detectable, and is added to the previous length estimate. During digitizing, decisions are made about data encoding precision. These decisions affect distance calculations.

Source material scale. The scale at which a map is compiled can influence length measurement because generalization, in the form of decreased line complexity, increases as map scales become smaller (Gardiner, 1982; Maling, 1968).

Source material error. Dueker (1975) has described error types that occur on source maps. If the source contains error, its digital product will, at best, replicate those errors with a high degree of fidelity. It is likely, however, that additional error will be introduced during data capture.

Error in data capture. When documents are captured in digital vector form, encoding errors occur (Dueker, 1975). Jenks (1981) has identified errors that occur during manual encoding of irregular linear features.

Error in data storage and processing. Chrisman (1984) argues for using good judgment in selecting the resolution for coordinate storage, and for the use of absolute, rather than incremental coordinates. Chrisman also describes other issues that must be weighed (e.g. real vs. integer, and decimal precision).

Topological error. Ginsburgh and Hansen (1974) describe three common error types in studies using road networks. Their third type deals with topological problems arising from incomplete specification of internodal linkages. In their framework the topology of a network is verified by reciprocal enumeration.

Aggregation error. Hillsman and Rhoda (1978), Goodchild (1979), and Casillas (1983) have shown that aggregating demand (e.g. households) to a single proximal point misrepresents the actual underlying distribution of demand and introduces error into analyses.

Also note that in some instances, other factors contribute to measurement error (e.g. document instability, projection). Each of the above factors can act alone, but error is often attributable to several factors acting in concert.

2.4 Distance Estimation

Many procedures used by decision support systems optimize distance-based objective functions. Researchers have suspected that simple interpoint distances are not always appropriate. As a result, locational analysis software (e.g. Goodchild and Noronha, 1983) often allows alternatives to symmetrical interpoint distances (e.g. time, or asymmetrical distance matrices to model one way streets).

Varying interpoint distances by a constant also has been the subject of empirical research because of differences between straight-line and network distances. Nordbeck (1963), for example, found that a constant (1.23) could be applied to straight-line distance to compensate for measurement problems. Love and Morris (1972; 1979) also examined several methods (e.g. Euclidean and spherical) for producing estimates, and compared each to 'actual' distances obtained from road map distance matrices and government agencies. The estimates were also parametrically altered. For example, simple Euclidean distances performed poorly when compared to estimates, such as:

$$D\ ij = 1.16*(\ \mathrm{abs}\ (xi-xj)\ **\ 1.60)\ **\ 1/1.62, \qquad [1]$$

where i and j represent two locations for coordinate pair x, and 1.16, 1.60, and 1.62 are estimated parameters.

Kolesar (1979) provides a distance approximation for deriving estimated travel times based on distances within U.S. urban areas:

$$D\ ij = 1.15 * \mathrm{SQRT}\ ((Xi-Xj)**2+(Yi-Yj)**2), \qquad [2]$$

where 1.15 is an empirical constant. The size and form of parameters change among these studies (e.g. 1.23; 1.16; 1.15), and Ginsburgh and Hansen (1974) assert that values ranging between Euclidean distance, and Euclidean distance * (1+SQRT(2)) are acceptable. This variability accentuates uncertainty in producing distance estimates, leads to problems with interpretation of results, and can lead to an inability to replicate spatial modeling experiments.

The entire estimation process may be obviated if 'actual' distances in the form of digital networks are available for analyses. Kolesar (1979), for example, suggests that using network models of actual street data to compute distances can yield accurate results. He does, however, note that 'extensive data base and time consuming calculations' are required (Kolesar, 1979). Information of this kind is available in the form of USGS Digital Line Graphs (DLG). These DLG data are an ideal source of reference data, because they are encoded at a high level of precision and are topologically validated (U.S.G.S., 1985).

3.0 METHODS

3.1 Study Area

A series of commonly used distance estimates are used to assess the magnitude of changes which occur in the locational analysis procedures. The site for the research is the Chickamauga 1:100,000 quadrangle in GA, AL, and TN. The site contains parallel ridges, and the road network, in part, reflects the topography; many roads running orthogonal to the ridges tend to be more sinuous than those running parallel to the ridges.

The quadrangle, which is a primary data source in the research, conforms to National Map Accuracy Standards: not more than 10% of points on the map are in error of more than 1/30 inch, measured at the publication scale (Thompson, 1979). This standard, however, applies to 'well-defined' points such as road intersections and benchmarks. Error in the trace of linear features, therefore, may be uncontrolled by the accuracy standard.

3.2 Demand Weights

A sample of 26 demand nodes was drawn from named places on the Chickamauga quadrangle, and if data were available, each node was assigned a population weight from the 1980 Census; for unincorporated places without available population figures, a weight derived from the mean of small places with known population, was assigned. An additional set of analyses was performed on the nodes with each weight set equal to one. The intent of this second set is to examine purely spatial effects on the outcome of the analyses.

3.3 Distance Calculations

The data sets, or estimation procedures used are:

1) Manually encoded network at 1:100,000;
2) Manually encoded network at 1:250,000;
3) Manually encoded network at 1:500,000;
4) Euclidean internode distance;
5) Manhattan internode distance;
6) Parameterized Euclidean distance (Equation 1).

Manual digitizing. Distances were measured using point sample digitizing on a series of USGS maps: Chickamauga, 1:100,000; Rome, 1:250,000; and the USGS state maps for Georgia and Alabama, 1:500,000.

Euclidean, Manhattan, and Parameterized. The location of each demand node was encoded, and a program calculated distances for each metric.

3.4 Data Analysis

The distributions were processed using a shortest-path algorithm and a vertex substitution method for solving the p-median problem (Tietz and Bart, 1968). For each analysis performed, the following questions were addressed: How does the value of the objective function change? Are different geometrical configurations produced when the distance measures are changed?

4.0 RESULTS

The analyses for the 26 nodes were performed with 10 sites to be assigned facilities. Each node was a site candidate.

4.1 Weighted Analyses

Objective function. The analyses for population-weighted data exhibited consistency across some of the distance measures (Table 1). For example, the results from the parametric, Manhattan, and the manually digitized networks at 1:100,000 and 1:250,000 have similar objective functions. The results, however, from the Euclidean and 1:500,000 digitized database, reflect large differences in the objective function.

DISTANCE METRIC	OBJECTIVE FUNCTION	% OF EUCLIDEAN
Euclidean	14,319.87	100
Manhattan	18,378.99	128
Parametric	17,134.81	120
1:100,000	17,678.11	124
1:250,000	17,629.29	123
1:500,000	20,463.39	143

Table 1. Results of weighted analyses.

Geometrical configuration. The 1:100,000 and 1:250,000 maps have identical configurations. The manually digitized map at 1:500,000, however, yielded a different configuration (Figure 1). This is clearly an inappropriate scale of analysis for the example problem, because of the abstraction of the road network. Additionally, the Euclidean and parametric analyses yielded identical configurations, that were different from the Manhattan or digitized results.

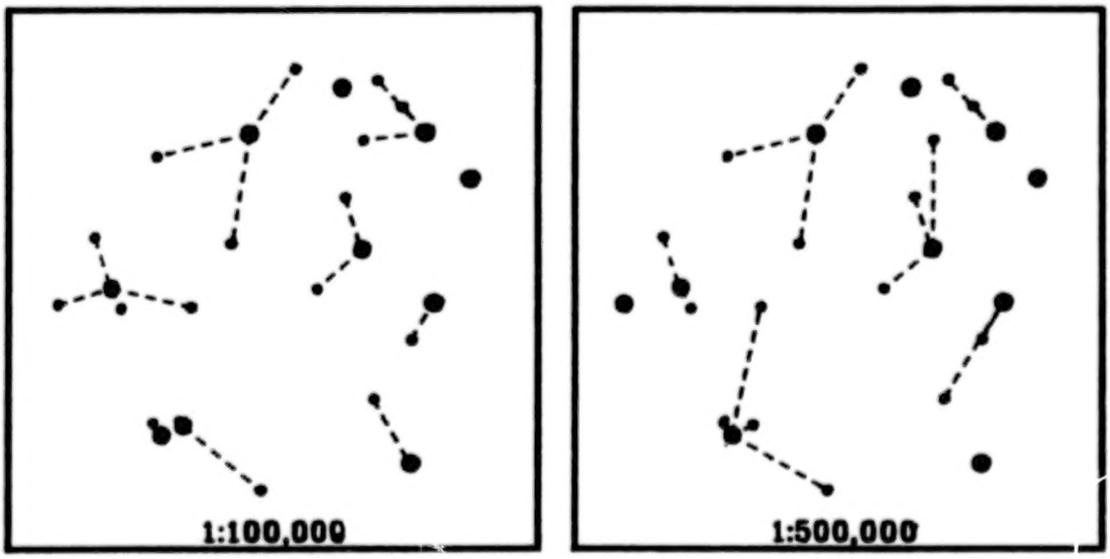


FIGURE 1. Geometrical configurations from weighted analysis

4.2 Unweighted Analyses

Objective function. The unweighted analyses reflect values attributable to distance variation. Although the objective function fluctuates slightly across analyses, it is notable that for the digitized series, it monotonically increases with the scale factor (Table 2). A similar pattern appeared in the weighted analyses (Table 1). Although the 1:100,000 and 1:250,000 analyses are reversed, their values are quite similar, and are much lower than the 1:500,000 results.

DISTANCE METRIC	OBJECTIVE FUNCTION	% OF EUCLIDEAN
Euclidean	92.38	100
Manhattan	120.74	131
Parametric	111.16	120
1:100,000	117.46	127
1:250,000	118.46	128
1:500,000	129.61	140

Table 2. Results of unweighted analyses.

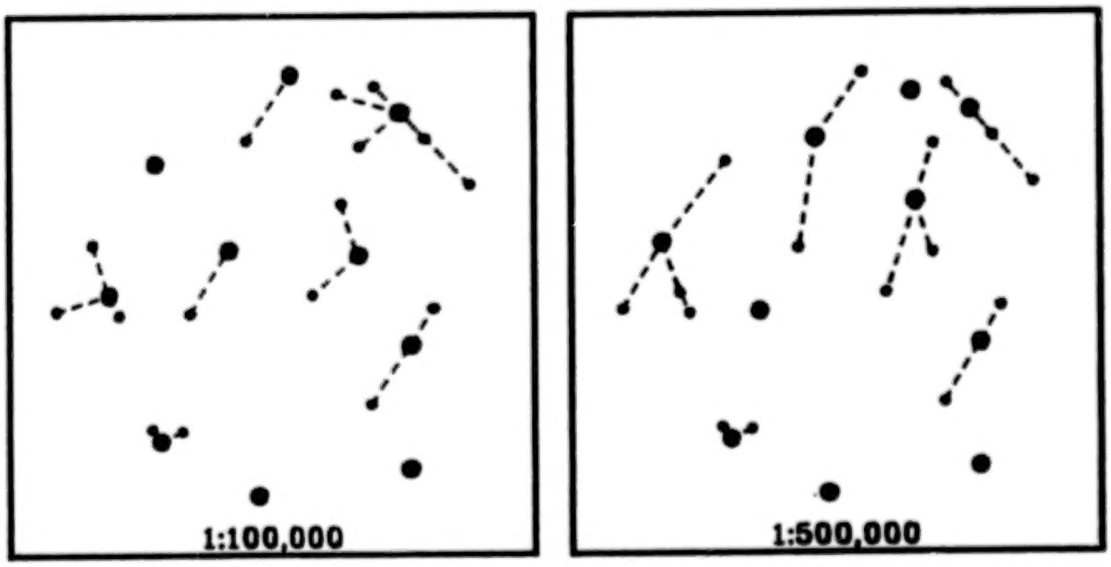


FIGURE 2. Geometrical configurations from unweighted analysis.

Geometrical configuration. The results exhibit a higher amount of instability than the weighted analyses. The Euclidean and parametric results are identical. The configuration, however, is different than that produced by the weighted analyses. The Manhattan metric produces a similar solution; it differs only because one node is assigned to a different center. Each of the digitized maps produced a unique configuration. Figure 2 shows a comparison of the 1:100,000 and 1:500,000 results.

5.0 CONCLUSIONS

This research contributes to knowledge about the workings of analysis procedures that will be used to assist decision makers in making judgments about important future issues. The main goal was to examine variations that result when different distance representations are used in locational models. The results indicate that outcomes are influenced by the scale of the source material or distance estimation procedure. Both the value of the objective function and the geometrical configuration of the allocation are altered when different, but commonly employed, distance estimation procedures are used in the analysis. An important effect centers on the extreme generalization of some data sources. Although generalization, and decreased sinuosity, of small scale maps are important in calculating distances, the decrease in distance is offset by another facet of generalization-- exclusion of low-order roads.

Future work will focus on use of existing digital data products. Most important is an examination of differences that may arise from the use of simple interpoint representations, digitized networks, and distance calculated from existing spatial databases (e.g. DLG).

ACKNOWLEDGEMENTS

The assistance of Panos Lolonis in data collection and processing is gratefully acknowledged. Paul J. Densham coded the location-allocation software and provided insight during the course of performing analyses. Jeff Hanson and David Bennet also provided assistance.

REFERENCES

Armstrong, M.P., Densham, P.J., and Rushton, G. 1986. Architecture for a microcomputer based spatial decision support system. Proceedings, Second International Symposium on Spatial Data Handling. Williamsville, NY: IGU Commission on Geographical Data Sensing and Processing.

Buttenfield, B. 1985. Treatment of the cartographic line. Cartographica 22(2), pp. 1-26.

Casillas, P. 1983. Data aggregation and the P-median problem in continuous space. Discussion Paper Number 35. Iowa City, IA: Department of Geography, University of Iowa.

Chrisman, N.R. 1982. A theory of cartographic error and its measurement in digital data bases. Proceedings, Auto-Carto 5. Falls Church, VA: American Society of Photogrammetry and American Congress on Surveying and Mapping.

Chrisman, N.R. 1984. On storage of coordinates in geographic information systems. Geo-Processing, 2, pp. 259-270.

Davis, J.R., and Grant, I.W. 1987. ADAPT: a knowledge-based decision support system for producing zoning schemes. Environment and Planning B, 14, pp. 53-66.

Densham, P.J. and Armstrong, M.P. 1987. A spatial decision support system for locational planning: design, implementation, and operation. Proceedings, Auto-Carto 8. Falls Church, VA: American Society for Photogrammetry and Remote Sensing.

Dueker, K.J. 1975. Geographic data encoding issues, Technical Report #43. Iowa City, IA: University of Iowa Institute of Urban and Regional Research.

Elmes, G.A. and Harris, T.M. 1986. Hierarchical data structures and planning optimization: an application to the Sussex land-use inventory. Proceedings, Second International Symposium on Spatial Data Handling. Williamsville, NY: IGU Commission on Geographical Data Sensing and Processing.

Falk, T.R. and Abler, R. 1980. Intercommunications, distance and geographical theory. Geografiska Annaler, B, 62, pp. 59- 67.

Gardiner, V. 1982. Stream networks and digital cartography. Cartographica, 19(2), pp. 38-44.

Ginsburgh, V. and Hansen, P. 1974. Procedures for the reduction of errors in road network data. Operational Research Quarterly, 25, pp. 321-322.

Goodchild, M.F. 1979. The aggregation problem in location-allocation. Geographical Analysis, 11, pp. 240-255.

Goodchild, M.F. and Noronha, V. 1983. Location-Allocation for Small Computers, Monograph 8, Dept. of Geography, University of Iowa, Iowa City, IA.

Hillsman, E.L. and Rhoda, R. 1978. Errors in measuring distances from populations to service centers. Annals of the Regional Science Association, 12, pp. 74-88.

Jenks, G.F. Lines, computers, and human frailties. Annals, Association of American Geographers, 71 (1).

Kolesar, P.J. 1979. Travel time and travel distance models. Ch. 6 in Fire Department Deployment Analysis. New York: North Holland.

Kuiper, J.H. 1986. Distribution of Distances in Pregeographical Space. Brookfield, VT: Gower Publishing Company.

Love, R.F. and Morris, J.G. 1972. Modelling inter-city road distances by mathematical functions. Operational Research Quarterly, 23, pp. 61-71.

Love, R.F. and Morris, J.G. 1979. Mathematical models of road travel distances. Management Science, 25 (3).

Lupien, A.E., Moreland, W.H., and Dangermond, J. 1987. Network analysis in geographic information systems. Photogrammetric Engineering and Remote Sensing, 53, pp. 1417-1422.

Maling, D.M. 1968. How long is a piece of string? The Cartographic Journal, 5, pp. 147-156.

Muller, J.-C. 1987. The concept of error in cartography. Cartographica, 24, pp. 1-15.

Nordbeck, S. 1963. Computing distances in road nets. Papers of the Regional Science Association, 12, pp. 207-220.

Nystuen, J.D. 1963. Identification of some fundamental spatial concepts. Papers of the Michigan Academy of Science, Arts, and Letters, 48, pp. 373-384.

Perkal, J. 1966. On the length of empirical curves. Discussion Paper Number 10. Ann Arbor, MI: Michigan Inter-University Community of Mathematical Geographers.

Peucker, T.K. 1975. A theory of the cartographic line. Proceedings, Auto-Carto II. Washington, DC: American Congress on Surveying and Mapping.

Peuquet, D.J. 1984. A conceptual framework and comparison of spatial data models. Cartographica, 21 (4), pp. 66-113.

Thompson, M. 1979. Maps for America. Washington, DC: U.S. Government Printing Office.

Tietz, M. and Bart, P. 1968. Heuristic methods for estimating the generalized vertex median of a weighted graph. Operations Research, 16, pp. 955-961.

U.S.G.S., 1985. Digital Line Graphs from 1:100,000 Scale Maps. Reston, VA: United States Geological Survey.

White, M.S. 1984. Technical requirements and standards for a multipurpose geographic data system. The American Cartographer, 11 (1), pp. 15-26.

BEST COPY AVAILABLE

CHOROPLETH MAPPING ON THE AMIGA

Cyrus B. Dawsey, III

Department of Geography

Auburn University, Auburn, Alabama

The computer revolution of the past decades has profoundly impacted one of the oldest professions: --- no, the other one; cartography. Mainframe systems of the late 1960's and early 1970's produced reams of printed maps for a variety of purposes. Already capable of manipulating large data files, researchers using SYMAP and other programs were able to rapidly and repeatedly inspect spatial patterns associated with many variables and areas.(1) Mainframe cartography had its drawbacks, however. Output was often crude, delays caused by the time sharing environment were common, and computers were affordable only to large organizations such as universities and corporations.

The arrival of inexpensive microcomputers in the 1980's provided map makers with new and challenging tools. Though total data storage capacity was less than present on a mainframe, they made available far greater flexibility and control to the end user. Microcomputer cartography, though, also presented problems. The generation of graphics was computationally intensive, so output on most machines was slow and restricted to a few colors and patterns. Specialized equipment such as plotters and digitizing tablets were relatively expensive, often costing more than the processor itself. Though commercially produced cartographic packages were available (software review is now a regular feature of The Professional Geographer), development did not match that of other applications such as spreadsheets and wordprocessors.(2)

Furthermore, a reliance on "canned" packages, in some ways, negated the advantages inherent in a microcomputing environment. Interaction was necessarily filtered through the programming of the original developer, flexibility was limited to those parameters which he/she deemed to be alterable, and input and output were restricted to high demand devices or patterns. The end user with a specialized need such as to map unusual data, collected from a rarely mapped area, to be displayed in a nonstandard way was left unsatisfied. Personal programming, moreover, was not usually feasible because BASIC was the language included with most computers and also the one with which most people were familiar. Though simple to use, BASIC, especially in the commonly bundled interpreted implementations, was slow and conducive to poor program organization.

Fortunately, the recent availability of speedier and more versatile machines may help computer cartographers overcome some of the difficulties mentioned above. The purpose of this report is to describe some probing of the potential for choropleth mapping on one of the newer microcomputers establishing a niche in the market: the Amiga by Commodore. Rapid and aesthetically pleasing map output has been achieved on a basic Amiga system configured without extra peripherals. For this study, input devices consisted of the standard keyboard and mouse, while output was to the RGB color monitor. Though some experimenting involved a commercial software painting package used to create outline basemaps, all programs were written in the Microsoft AmigaBASIC interpreted BASIC which is provided with the machine.

1. THE AMIGA

The introduction of the Amiga 1000 in 1985 generated much excitement in the microcomputer world. Reviewers described it as a landmark machine with capabilities unavailable in any other comparably priced ($1000 - $2000) system.(3) Its sound chip could produce music in stereo and talk in English or another language; its fast Motorola 68000 processor provided true multitasking by running applications independently of each other; and its architecture was open for easy expansion. The most attractive feature of the Amiga, however, was the ability to produce colorful graphics. Dealers fondly displayed bouncing balls and multi-hued robots to demonstrate the selection pallette of over 4000 colors though only 32 could be easily displayed on a screen resolved at 320 x 200 pixels. Half as many colors could be simultaneously shown at a resolution of 640 x 400. Furthermore, because the hardware was designed to minimize the time required for generating graphic output, convincing animation effects were possible. While written line commands provided complete access to DOS, a mouse controlled environment of icons and windows was created to be the primary user interface.

Microsoft's BASIC included with the Amiga was an extensive implementation of the language, and though an interpreted version, it was faster than BASIC on many other systems. Furthermore, several graphic commands exercised direct hardware control through specialized chips thus bypassing the central processor. This added to the overall speed.

2. PROGRAM DEVELOPMENT

Choropleth mapping entails the development of 3 basic elements; the data values to be mapped, basemap information, and instructions on how to produce the map. Exploration of the Amiga's capabilities involved an examination of several options for basemap development as well as extensive programming for producing final output maps. On the other hand, the creation of the first and perhaps most important element, a data base of the variables to be mapped, was simple and straightforward. The number of area units, their names, and the associated data values were stored in a sequential file. Data input to the file was facilitated by a program which displayed an outline map successively highlighting each area unit while requesting input of data values associated with that unit.

2.1 Basemap Digitizing

Several options were considered for developing basemap files. Commercially available lists of digitized coordinate values for county, state, nation, or other standard map boundaries can be purchased; and though not available directly for the Amiga, they might have been transferred from other formats. The objective here, however, was to produce maps from scratch as might be necessary in a study of a remote and forgotten corner of the globe. A digitizer program was written which made use of the mouse, standard equipment on the Amiga, as the primary input device. Digitizing was accomplished by fixing a transparent sheet, on which a map was depicted, over the monitor. The cursor was then positioned on the screen at boundary intersection points or border direction change points, and each was stored as an x-y coordinate of a line segment endpoint by clicking the mouse button. Because the coordinate pairs represented vertices of the basemap outline, each pair was contained in two or more of the line segments. Keyboard controls allowed for "grabbing" a vertex by outlining a small rectangle on the screen movable with the mouse. When a vertex was brought within the rectangle, its coordinate values could be established as the end of the current line segment or the start of a new one. A similar "capture" method allowed for line erasure. Though the procedure involved intensive BASIC looping, time delay was within acceptable limits. Digitizing maps of Alabama, the United States, Brazil, and Africa, each containing several hundreds of line segments, was accomplished in less than 15 minutes per map. As a final step in the digitizing procedure, an estimated centroid point in each area unit as well as its name were identified. Coordinates for the centroid points were necessary for later use of the BASIC paint command which fills an enclosed area of the screen. A final step wrote all files to disk. Though use of a digitizing tablet or pre-developed coordinate files might have made possible greater accuracy, the method used did provide what was judged to be adequate outline maps useful for pattern analysis.

2.2 Choropleth Mapping

The third step in developing a choropleth mapping package consisted of writing a "mapmaker" program through which a user could actually produce a map. The developed program included a menu driven main loop which called several subroutines to accomplish the following:

1. Read in a basemap file.
2. Read a data file.
3. Compute value range interval categories.
4. Set colors to the interval categories.
5. Draw the outline map.
6. Paint each area unit with the proper color.
7. Quit and close all files.

Steps 1 and 2 were straightforward disk file operations. The identification of the value range intervals in step 3 involved inputting the number of categories (2 to 14) and then selecting a "manual" or "automatic" option. "Manual" allowed the user to identify the value range break points while "automatic" caused a programmed routine to develop the categories. The "automatic" algorithm compensated for skewed data and attempted to include approximately the same number of observations within each of the intervals. Top and bottom interval ranges (i.e. from the minimum value to the lowest break point and from the maximum value to the highest break point) were allowed to be of variable width and different from that of the standard width of the other interval ranges.

Color selection (step 4) was from a pre-established gradient of light to dark shades in a red hue. Once mapped, the hue base could be changed to blue or green or back to red. Following initial access of the basemap from disk and storage as an array in memory, the outline map could be drawn (step 5) almost instantly by looping through BASIC's line command. Choropleth mapping occurred as the program painted each area unit with a color value matched to the proper value range interval. The painting was performed by calls to the BASIC paint command which colored in an area of the screen starting from the pre-defined centroid point to the area unit boundary lines.

The whole procedure could be repeated by issuing appropriate keyboard commands in response to menu requests. The value ranges might be reset any number of times, using any number of categories, or a completely new data file and/or map file might be brought up without ever exiting the program.

As mentioned above, the basemap development required several minutes, but the "mapmaker" program ran very quickly. Once the basemap file had been produced, therefore, very little time was used in making the choropleth maps. The outline map of Brazil and its 26 states, for example, could be displayed in 3.1 seconds and completely colored in an additional 8.6 seconds.

2.3 Delux Paint Maps

A second method for developing basemaps was explored. Because the final choropleth patterns were produced by painting an area of background color enclosed by a foreground colored outline, a basemap might be created by any procedure capable of drawing a map on the screen. The Delux Paint commercial program (under $100.00) released for the Amiga by Electronic Arts is a widely distributed general purpose collection of painting and drawing routines. Many options including smear, shade, line, outline, brush capture, palette change, and perspective drawing to list only a few, are available for the user. Amiga oriented computer magazines often include impressive artwork produced using Delux Paint.(4)

Outline maps were drawn with various embellishments such as three dimensional perspective or professional font labeling and stored in a file. The Delux Paint file was then converted to an array which was readable in BASIC. Though the conversion process was somewhat tedious and involved 3 separate steps, the final form Delux Paint derived basemaps could be used and filled in as easily as the line segment outline maps developed previously. Again, all programs were written in AmigaBASIC. The use of Delux Paint made possible the generation of basemaps with a finished and professional quality suitable for hardcopy or videotape presentation.

2.4 Polygon Maps

A third method of choropleth mapping was investigated. The commands area and areafill are included in the AmigaBASIC version of the language. These allowed for the description of a polygon defined in an array of up to 20 successive outline coordinate points and for the coloring in of the identified area. Execution of these commands was more directly implemented in the hardware and, therefore, more rapid than the paint command used previously.

A program was written which converted the outline maps, made up of not necessarily consecutive line segments, to a group of polygon outlines described by ordered vertices. If polygons included more than 20 vertices, they were split and stored as 2 or more separate units to be merged in the mapping program. Except for the external boundary, all sides of the polygons matched exactly with one other polygon side.

A slightly modified "mapmaker" program was created to match the polygons to data values and then display them in the proper color by calling the areafill command. Drawing an outline to the screen first, previously necessary for use of the paint command, was no longer needed. The overall speed of map display was increased to 3.6 seconds for the full map of Brazil. This was considerably faster than the 11.7 seconds required when the paint command had been used to fill in a basemap of line segments or painted outline.

3. CONCLUSION

Choropleth mapping using the areafill command was almost rapid enough to permit convincing representation of pattern shifting through time, or other pseudo-motion effects. Though not attempted, double buffering and other enhancements might have provided smooth animation of a diffusion process, commodity flow, or simulated land use change. While animation is not uncommon in computing, it is elsewhere accomplished on larger machines using more powerful compiled languages. The Amiga was shown to make available interactive cartography and spatial analysis to an ordinary user familiar with no more than BASIC, and it hints at far more.

Neither the limits of the machine nor the AmigaBASIC language were achieved by the programming described above. Polygon mapping using a Delux Paint background, for example, was not attempted, nor were special effects such as 3 dimensional rotations tried. The preliminary probings of the Amiga and its accompanying version of BASIC, however, do suggest that the computer has great potential as a cartographic tool. While other systems may provide more and better output, the Amiga brings computer mapping applications to people who would not otherwise have access. It, therefore, stands to make a significant contribution to the field.

References

1. FRASER TAYLOR, D. R., "The Computer in Contemporary Cartography," FRASER TAYLOR, D. R., ed., Progress in Contemporary Cartography, New York: J. Wiley, 1980, p. 6; CARTER, J. R., Computer Mapping, Washington: AAG, 1984, pp. 44-45; SYMAP is a product of the Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University.

2. See for example: The Professional Geography, Vol. 39(3), August, 1987, pp. 357-362.

3. WILLIAMS, G., EDWARDS, J.: AND ROBINSON, P.; "The Amiga," Byte, Vol. 10(8), August 1985, pp. 83-100; ANDERSON, J., "Amiga," Creative Computing, Vol. 11(9), September 1985, pp. 32-41.

4. All issues of Amiga World, for example.

Evaluation of the 1:250,000 Digital Elevation Model for Use in a GIS for the Great Smoky Mountains National Park

James R. Carter, Ph.D.
Associate Professor, Geography, and
Associate Director, Computing Center
University of Tennessee
Knoxville, TN 37996

ABSTRACT

Because exisiting 1:24,000 DEMs cover less than 40% of the area of the Great Smoky Mountains National Park a tape of the 1:250,000 DEM covering the entire Park was purchased and evaluated for use in GIS investigations by environmental scientists working on the Park. This smaller-scale DEM displays the inherent generalization of the map from which it was derived, but otherwise the DEM seems to be as accurate as the map from which it was digitized.

To keep from having to fit the Park boundary to the large database everytime it is accessed and to cut down on the size of the database being read, a revised DEM was created. The database was cut in half by eliminating all of the rows of elevations well to the north and south of the area of the Park. The revised DEM is still rectangular and data are stored in the same format as in the original DEM. In the new database all elevations that were found to be outside of the Park were multiplied by -1.0. This revised database has proven to be useful and easy to work with.

A program was written to map slope by categories for the entire Park. Two slope maps are included in this paper. Analysis of the slope maps has given new insights into patterns of the topography of this complex area. Summary statistics concerning various frequencies of slope and elevation were accumulated based on the 295,000 elevations that were found to fall within the Park. In another computation it was found that the actual surface area of the Park is only 2.4% larger than the planimetric area of thePark.

Although purchased as a surrogate for the larger-scale DEM data, the 1:250,000 DEM has proven to be a useful database for what it can show about the Park at its own scale. There is still a need for the larger-scale DEM data, but that data will be used in addition to the 1:250,000 DEM data, not in place of it.

1. INTRODUCTION

The Great Smoky Mountains National Park scientific staff has been building a Geographic Information System to address many environmental questions and to create a map of vegetation. In the process the need for various topographic parameters has been identified to assist in addressing many of the environmental questions. This need has lead to an interest in using Digital Elevation Models. About one-third of the area of the Park is covered by 1:24,000 DEMs and all of these are early products of the USGS Digital Cartographic Program, having been created on the Gestalt PhotoMapper system and having had minimal editing (Carter, 1984). DEMs for the rest of the quads covering the Park are supposedly in the works and have been forthcoming for years. In lieu of waiting until the larger-scale DEMs became available, the Western 1/2 of the Knoxville 1:250,000 DEM was purchased for evaluation and possible use in selected studies. This one model covers more than 99.9% of the area covered by the Park. No thought was given to purchasing data to cover the tip at the west end of the Park that extends on to the neighboring 1:250,000 map.

2. EVALUATING THE 1:250,000 DEM

It was understood at the beginning that the 1:250,000 DEM could be no more accurate than the map from which it was derived and that this map is quite generalized as is appropriate for a map of this scale. The 1:250,000 DEMs contain elevations in the 3 arc-second format. At the middle latitude of the Park, 35.7 degrees north, the north/south spacing between elevations is 92.4 meters and the east/west spacing is 75.6 meters. The author made preliminary tests of the quality of the DEM by pulling rectangular matrices of selected areas and generating contour maps of the matrices (Carter, 1983). These contour patches were compared with the 1:250,000 map and the 1:24,000 topographic quadrangles. In all cases the patches showed that the data was a good representation of the contour pattern portrayed on the smaller scale map. Except in areas of complex local relief, the matrices from the 1:250,000 DEM captured the general topographic patterns as shown on the larger-scale 1:24,000 DEMs. Although it has been reported that there are serious error problems with a few of the 1:250,000 DEMs, there appears to be no reason to suspect any systematic errors with this particular model.

To evaluate further how accurate this DEM is and to determine what this information could contribute to the GIS, the author chose to use the DEM to make a map of slope for the Park. There is a separate 1:125,000 topographic map of the Park, also available in a shaded relief edition (U.S.Geological Survey, 1972). It was decided to make the slope map at this 1:125,000 scale so one map could be laid over the other to help in identifications and to make comparisons.

3. BUILDING A SLOPE MAP

The first concern was to define the area of the Park and fit a boundary to the DEM elevations. The author and the staff of the Smokies did not have available a GIS package or mapping package that would permit handling files of this size or studies of this extent. Therefore, the author wrote all of the software, building upon existing programs at hand. The boundary of the Park was digitized from the 1:125,000 map and fit to the DEM. Because the Park extends only 23 minutes north to south it was decided to treat the map of the Park as a simple cylindrical projection with the length of an unit of longitude reduced by the cosine of the latitude of the middle of the Great Smoky Mountains National Park. This simple type of projection can be used because with the little latitudinal extent the variation in horizontal spacing between sample elevations from the average within the Park is less than 0.3%. Because the Polyconic map projection on the 1:125,000 map displays such little curvature across the width of the Park, there was no need to transform the digitized boundary to fit the cylindrical projection.

In the full 1 degree by 1 degree DEM there are 1201 X 1201 = 1,442,401 elevations. To save the process of fitting the boundary to the matrix each time data on the Park was extracted and to save reading an excessive number of elevations that are outside of the Park, a new matrix of elevations was created specific to the Park. The new matrix was written with 600 elevations south to north which was adequate to cover the Park and still give substantial overlap. The full 1201 columns of elevations were repeated in the new matrix. To identify the area within the Park the digitized boundary of the Park was fit to the DEM database by converting coordinates to row and column positions. In creating the new DEM all elevations outside of the Park were multiplied by -1.0 before they were written to tape. Because all elevations in the area are well above sea level, there is no basis to confuse a negative elevation with an actual elevation. By flagging all elevations outside of the Park with a negative sign it is easy to identify the area of the Park, and by using absolute values, elevations inside and outside of the Park can be used in calculations with equal ease.

The only plotter available to the author to portray the map was a CalComp 4-pen plotter normally mounted with black, red, green and blue pens. To get other colors requires special setups and considerable delays. Because these default colors do not form a natural gradation a set of density shades in a single color was sought out. But at the scale of 125,000 it was found that the distance between elevations was 0.0238 inch horizontally and 0.0293 vertically and because the normal width of a line on the plotter is only about one-half the horizontal spacing between elevations, only two symbols could be used for a gradation: a single vertical line and a pair of vertical lines which usually come out as a solid cell. Thus, with a single color three categories could be portrayed by using an open area for one category, an intermediate pattern consisting of the single vertical line, and a dark pattern consisting of two parallel lines that normally bleed together to form a solid pattern (Figures 1 and 2). By using green to represent two categories of low slopes and red for two categories of steep slopes effective maps with five classes have been produced using white space to represent the intermediate category. When reproduced in black and white only three classes can be portrayed.

Slope and aspect were computed using an algorithm developed by the author. This algorithm uses the 'spatial derivative' (Tobler, 1970) to calculate slope but what is unique is that the derivative is calculated using the four corners of a rectangular cell. Because in the case of the arc-second DEMs these four corners do not form a square the given elevations have to be extended to form a square. Based on the four corners of a cell the resulting slope and aspect are applied to the center of the cell, rather than to any of the original data points. This approach to calculating slope and aspect gives a value based on a sample area only one-half the size of the more conventional approach of using the elevations north, south, east and west of the elevation in the center. In the more conventional approach, the elevation in the center is neglected in the computation of slope and aspect. In this mapping program aspect is calculated in a conventional manner (Burrough, 1986, 50) but the results have to be rotated by 45 degrees.

The resulting slope map has been quite revealing because it gives a global view of the Park (figures 1 and 2). It shows that the eastern half of the Park is much steeper than the western half of the Park. Researchers who are familiar with the Park have commented upon examining the map that they see patterns that they had not seen before but that fit with what they know about the Park. Basically, then, a map of slope at this scale gives a perspective that is unique and for this reason alone such a map is valuable. Another pattern that has been revealed by this series of maps is that ridgelines and stream courses stand out as ribbons of relative flatness on the map (figure 1). As should be expected, in most cases the ridgelines stand perpendicular to the streamlines.

There is no conventional way to show aspect for the entire Park, other than by incorporating aspect into shaded relief models of the Park. Compilations of the frequency of occurrence of aspect have been collected for all of the 295,000

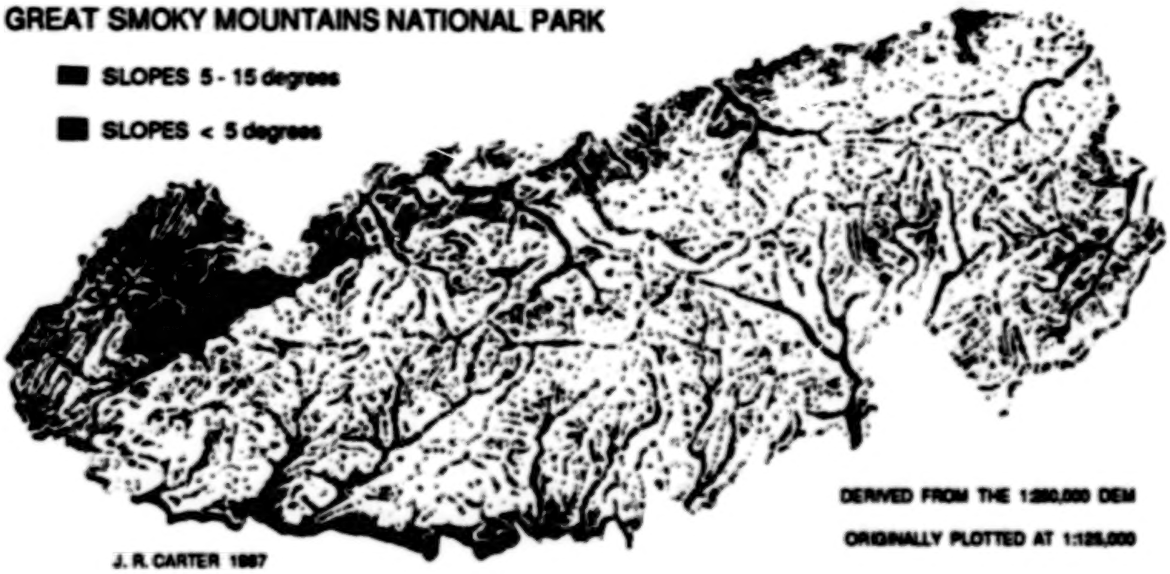

GREAT SMOKY MOUNTAINS NATIONAL PARK
■ SLOPES 5 - 15 degrees
■ SLOPES < 5 degrees
DERIVED FROM THE 1:250,000 DEM
ORIGINALLY PLOTTED AT 1:125,000
J. R. CARTER 1987

Figure 1 - Slope map of the Great Smoky Mountains National Park showing the areas of low slopes. More than 10% of the Park falls into the lowest class with slopes of less than 5 degrees. The map is based on the computation of slopes for more than 295,000 cells within the Park.

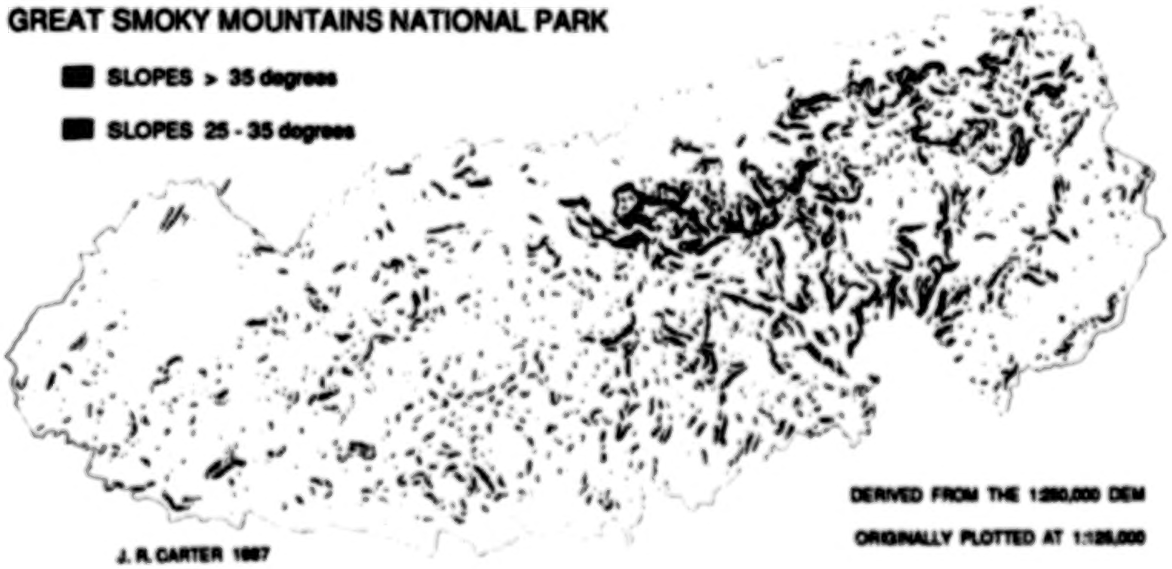
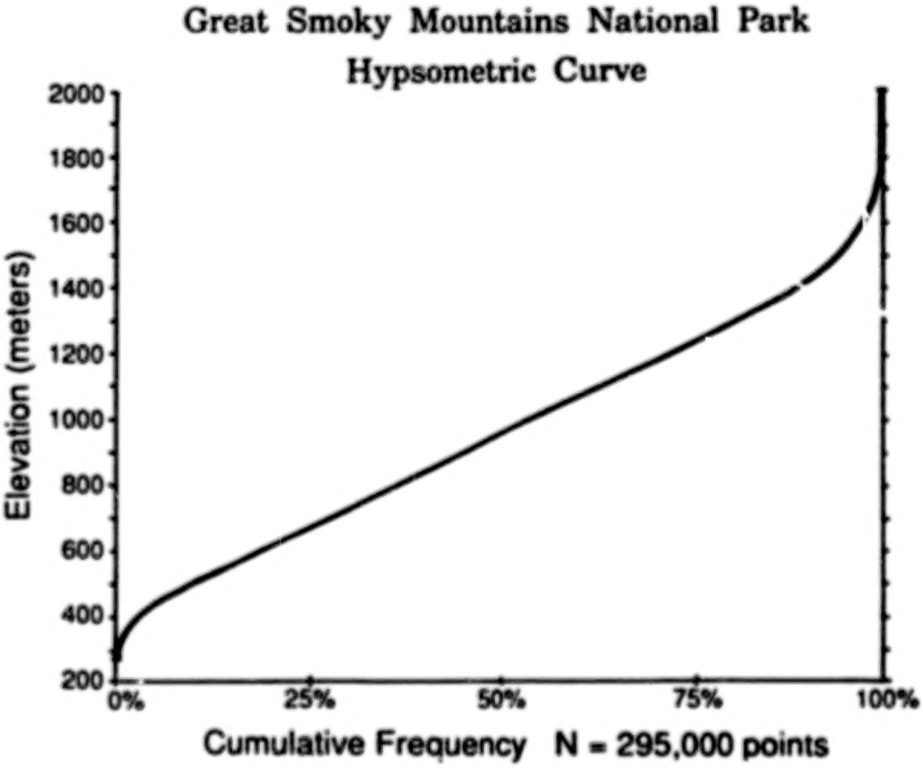

GREAT SMOKY MOUNTAINS NATIONAL PARK
■ SLOPES > 35 degrees
■ SLOPES 25 - 35 degrees
DERIVED FROM THE 1:250,000 DEM
ORIGINALLY PLOTTED AT 1:125,000
J. R. CARTER 1987

Figure 2 - Slope map of the Great Smoky Mountains National Park showing the areas of steep slopes. Only 10% of the Park has slopes greater than 25 degrees. The map is based on the computation of slopes for more than 295,000 cells within the Park.

sample sites. These aspect values have also been aggregated by various slope classes. To this point nothing of value has been gained by looking at aspect statistically. Aspect will be an important component in the analysis of the vegetation pattern.

4. STATISTICAL TABULATIONS

Some informative statistics concerning slope and elevation have been derived from this DEM database. By counting the number of cells falling into elevation classes in 100 meter increments it was possible to create a graph of the cumulative area as a function of elevation -- what is sometimes called the Hypsometric or Hypsographic Curve. Figure 3 shows this relationship for the Great Smoky Mountains National Park. Note how linear the relationship is between 600 meters and 1300 meters. Within this category the most any single category departs from the mean is by 11%.


Great Smoky Mountains National Park

Hypsometric Curve

Figure 3 - Curve showing the relative amount of the area of the Great Smoky Mountains National Park as a function of elevation. The curve is based on more than 295,000 elevations from the W 1/2 Knoxville 1:250,000 DEM.

The modal slope of the 295,000 cells is about 13.5 degrees. The median slope is slightly less than 13.0 degrees. About 4% of the cells have a slope of less than 1.0 degree and the maximum slope calculated is 65 degrees. Only 2% of the cells have a slope of 45 degrees or greater. Breaking the slope data into deciles reveals that 50% of the Park consists of slopes in the range of 10 to 20 degrees.

10% of the Park has slopes less than 4 degrees
20% of the Park has slopes less than 8 degrees
30% of the Park has slopes less than 10 degrees
40% of the Park has slopes less than 12 degrees
50% of the Park has slopes less than 14 degrees
60% of the Park has slopes less than 16 degrees
70% of the Park has slopes less than 18 degrees
80% of the Park has slopes less than 20 degrees
90% of the Park has slopes less than 25 degrees

One of the more interesting questions that could be answered by examination of this data is how the actual surface area of the Park compares with the planimetric area of the Park, or put another way, 'how much larger would the Park be if it could be laid out flat?' Rather than basing such a calculation on an average slope statistic for the Park, the proportional increase in surface area of each cell was calculated by taking the secant of the slope of the patch. The proportional areas of each patch were added together to get the total increase in area. This calculation showed that, based on the 1:250,000 DEM database, the actual surface area of the Great Smoky Mountains National Park is 2.4% larger than the planimetric area of the Park. This increase in area is much smaller than what most people guess it would be. But, since the hypotenuse of a 12 degree slope is only 2.2% longer than the base, a 2.4% increase for the entire Park is in line.

It should be noted that there is some smoothing inherent in any gridded DEM because vertical and near-vertical slopes are lost in the spatial sample. Thus, this 2.4% increase in area should be qualified by noting it was calculated using the 1:250,000 gridded DEMs.

5. OTHER APPLICATIONS

This revised elevation database has been used by other researchers developing a vegetation map of the Park based on Landsat multispectral scans taken at two different dates. Still to be resolved is one of the original questions which prompted the work on setting up a GIS. The European Wild Boar has been introducted into the Park and has become a pest largely because of its widespread disturbance of soils with its practice of rooting. A removal program is in operation but it has been difficult to evaluate how effective the program is or where to concentrate removal activities. There are many reports of sightings of disturbed soils but it has yet to be determined if the spatial pattern of these sightings is reflective of the range of the Hogs or whether the sightings are highly biased toward the areas where people hike and thus come across areas of disturbed ground. Using the 1:250,000 DEM data, a statistical analysis of the nature of the Hog rooting in neighboring watersheds in currently underway.

When the vegetation classification is complete the vegetation data will be combined with data on elevation, slope and aspect to determine the typical areas where Hogs are found to root. If it is found that there are areas away from the trails that suffer the same damage as known rooting areas, then it may be appropriate to modify the operation of the removal program. The scale of this investigation will benefit from use of the larger-scale DEMs, for the 1:250,000 DEM database will probably be found to be too general to be integrated with the detailed network of trails.

6. CONCLUSIONS

This 1:250,000 DEM seems to be an accurate representation of the map from which it was derived and after much use of the data there has been no indication of any obvious or significant errors. While originally employed as a surrogate for the larger-scaled DEMs, this 1:250,000 DEM has proven to be a useful database in and of itself because it gives a different perspective of this topographically diverse area. There is still a desire to get complete DEM coverage of the Park at the 1:24,000 scale but even with the larger-scale data there will continue to be reason to refer to the 1:250,000 DEM database.

Acknowledgements

Credits are due to the University of Tennessee Computing Center, the University of Tennessee Geography Department, and the National Park Service, Great Smoky Mountains National Park, Uplands Research Lab for the support of this work.

References

Burrough, P. A., 1986, *Principles of Geographical InformationSystems for Land Resources Assessment*. Oxford: Clarendon Press.

Carter, James R., 1983, "Bringing the Digital Elevation Model into the Classroom," *Technical Papers, American Congress on Surveying and Mapping, 43rd Annual Meeting*, pp. 474-482.

Carter, James R., 1984, *Computer Mapping: Progress in the '80s*. Association of American Geographers Resource Paper.

Tobler, Waldo R. (ed.), 1970, *Selected Computer Programs*. Ann Arbor: Department of Geography, University of Michigan.

U. S. Geological Survey, 1972, *Great Smoky Mountains National Park and Vicinity*. Map at 1:125,000. 1972 Revised Edition of the 1949 Base Map.

BLANK PAGE

MODELS

BLANK PAGE

THE MATURING OF GIS AND A NEW AGE FOR GEOGRAPHIC INFORMATION MODELING (GIMS)

Jack Dangermond
Environmental Systems Research Institute
380 New York Street
Redlands, CA 92373

ABSTRACT

This paper begins with a brief review of the recent history of the development of Geographic Information Systems (GIS's), of the need to apply modeling to world problems, and suggests that the technology of Geographic Information Modeling Systems (GIMS's) is about to emerge. After drawing distinctions between modeling and the other kinds of data manipulation performed by GIS's, a list of models which work with GIS's and which are likely to be valuable to users is presented. Then four categories of modeling activity are recognized: static, deterministic modeling; simulation modeling; modeling systems interactions and modeling in a decision analysis environment. Finally the present climate and the future prospects for the development of GIMS technology are assessed, with special attention to the need for better GIS databases to support such development.

1. THE EMERGENCE OF THE GEOGRAPHIC INFORMATION SYSTEM

During the past 20 years, we have seen the emergence of a new concept in information handling, the Geographic Information System (GIS), dealing with a special type of information, spatial data. This information handling has required the development of new data structures to handle not only efficient organization and storage of cartographic data and their attributes, but also the various topological relationships among these cartographic information sets. The application of relational DBMS concepts and graph theory for supporting relationships among map features, and the development of hundreds of new spatial operators for entry, management, analysis and display of these geographic features has emerged to meet these needs. This software technology is beginning to stabilize and beginning to be acknowledged as a valuable tool for the sciences, decision analysis and making organizations run more efficiently.

This trend has had many useful by-products: GIS's tend to follow the principles of modern computer science and computer management; for example, redundancy is minimized and users have fast access pathways to information through interactive queries, graphics and analytic functions.

During this time of rapid development, among theoreticians as well as users of the technology the idea has slowly evolved that the development of the GIS, while very useful as a tool for managing spatial data and enhancing productivity, can also be a means to a further end: it is potentially useful for strategic analysis and decision analysis.

2. THE NEED TO APPLY MODELING TO WORLD PROBLEMS

At present there are a whole series of pressing problems facing the world. The human population continues to grow, tropical forests are rapidly being cut down, the demand for energy is difficult to meet, tensions over international trade are increasing, etc. Most of these problems involve complex natural and human-made systems, often intertwined. Such complex systems are very difficult to understand and predict. One approach to dealing with such systems is the use of various kinds of models, often executed on a computer and just now beginning to involve the use of geographic information. Such models, properly designed, can be powerful analytical tools, and may provide the most inexpensive means available to us for understanding these systems and for working with them. Lacking such models we may be forced to continue working with these systems based either on past experience, which is often misleading, or based on "experimenting" to see what will happen in each new situation: these are both increasingly risky approaches to dealing with the problems facing us.

Given the needs of the world and the increasingly sophisticated and powerful computer, GIS and modeling tools available to us and the rapid growth of geographic databases, it is likely that modeling will come to be much more widely used in the near future.

3. THE COMING EMERGENCE OF GEOGRAPHIC INFORMATION MODELING SYSTEMS (GIMS's)

As a result of this thinking, we are about to begin another generation of tool building: the development of the Geographic Information Modeling Systems (GIMS), built upon the base of existing GIS technology.

We have had between twenty and thirty years of experimentation with digital modeling tools for spatial information, but most of these experiments have been isolated projects. Now emerging is a trend toward the use of such models by long-term users of spatial information. Over the next decade we'll see increasing use made of the many models pioneered in geography, but they'll be executed using software engineering tools connected to a GIS. Those geographic information systems built around the DBMS concept will pipeline information to these models and interact with them most effectively.

4. THE DISTINCTION BETWEEN MODELING AND OTHER GIS OPERATIONS

In general, modeling is simply an attempt to represent one thing in terms of another. Often something real is represented in terms of something that is abstract; for example, spatial realities may be represented mathematically, logically, graphically, etc.. Of course, every map is thus a static model of geographic reality. Each of the mapped variables represents a corresponding real feature which is spatially distributed; vegetation polygons, for example, represent a model of the actual distribution of various kinds of plants over the landscape. The form of models using geographic information can range from the very simple to the very sophisticated. For ancient voyagers a navigator's model of Pacific islands was made with twigs bound together; in recent years we have moved from paper maps to digital data stored in GIS's.

A GIS is basically a system for manipulating digitally represented geographic information, that is, for manipulating one kind of a model of geography. But manipulating a model is not the same as creating a model or making a model work. Manipulations merely provide, in effect, a "different view" of the same thing. Modeling involves something new. This discussion will distinguish between those operations which are largely within the domain of geographic information systems (e.g., the manipulation of spatial data, building relationships between spatial data, etc.) and the modeling process which uses those GIS operations. For example, modeling with spatial information differs from simply building relationships among and between various spatial data sets. Polygon overlaying, for example, is not modeling; it builds relationships between two coverages (map layers) of information and builds tables of the interrelated attribute information which result from the overlay of the maps. Subsequent to the relationships being built, one can model with them. This modeling can be conceptually considered as a "different view" of the information already on hand. Thus, for example, the overlay of soils and vegetation polygons using a polygon overlay spatial operator results in joining the soil and vegetation tables in such a way that arithmetic and logic can be applied to them in order to interpret their interrelationships. (Polygon overlaying can thus be thought of as one way of building relationships between tables of data.) Other GIS operators, such as point-in-polygon and line-in-polygon build spatial relationships in a data base for supporting similar arithmetic and logical manipulations. Similarly other operations such as distance buffer, measuring vertical distance and address geocoding/aggregation could all be thought of as merely tools that build relationships among and between spatial features. These relationships are fundamental for modeling.

5. MODELS WHICH WORK WITH GIS's

Which kinds of geographic or spatial information modeling have already proved their usefulness? Which will work with present GIS's? Which are likely to be used with the coming GIMS's? The following summary is written for users; while not exhaustive, it will provide some idea of the breadth of such models and their potential usefulness in applications.

5.1 Reclassifications

One of the simplest kinds of spatial modeling involves the reclassification or reinterpretation of mapped variables. For example, soils data may be reinterpreted, reclassified or "modeled" to provide interpretive estimates of soil erosion potential or suitability for various crops. This procedure is used to predict future soil erosion or crop success by modeling a presumed relationship between the soils and a particular process such as erosion or the process of growing certain crops on them. Similar reclassification can be done on lines and points, by grouping attribute classes and resymbolizing the basic geographic features. Typically users consider this type of map manipulation to be a form of interpretive mapping. It also normally invokes some forms of feature generalization (e.g., polygon dissolve).

5.2 Logical and Arithmetic Tables

This kind of modeling involves the application of logical and arithmetic operations to map features with multiple attributes which are held in a tabular form. Using such logical and arithmetic models some form of tables, for example, urban sociologists often use census information to provide estimates of social status, based on the normalization and summarization of income, education and percentage of white collar workers. Similar types of logical and arithmetic manipulations have also been done in the natural sciences, land use planning and environmental assessment. These disciplines make use of decision formulae and decision tables to estimate the areas that are suitable for certain kinds of land uses as a function of variables such as soils, geology, rainfall, slope and existing land use. A GIS system is normally used to build all the relationships among and between these variables using tools such as ARC/INFO.

5.3 Interpolation

Another static modeling tool involves the creation of theoretical surfaces based on randomly distributed spatial data. One can model a surface from a set of single data points by using the mathematics of interpolation; one creates a new, three-dimensional surface of spatial reality. Both cellular as well as triangulated irregular network (TIN) techniques support such modeling. Such models provide measurements of surface value and distribution and can be visualized as a spatial pattern between and among the set of individual information points. Contoured displays, 3-D displays, displays of cross section analysis, etc., are not models, but rather visualizations of the new, interpolated surface.

5.4 Thiessen Models

Very similar to the interpolation models are Thiessen models, in which point information is modeled into a series of polygon definitions, using nearest neighbor concepts. That is, polygons are automatically generated which surround point phenomena and which separate the vectors which connect the points.

5.5 Flow Models

Another category of widely used spatial models deals with flows. Some of the best examples are from water resources. The HEC models for watershed simulation deal with flows through a network. They are based on a spatial network defined using a geographic information system. For example, rain falls in one location, is allocated to watersheds, then to various river segments (with varying impedances) in the overall hydrologic network, ultimately discharging into the ocean. This flow simulation requires the construction of a complex set of explicit relationships and involves modeling of run-off, etc. Flow models are usually quite sophisticated because they involve relationships in both space and time.

5.6 Meteorological Models

Diffusion models have been used chiefly in analyzing air pollution, where stacks or other point sources give off measured amounts of contaminants or particulates and where, within a cellular GIS environment, these materials are diffused through space (given defined wind speeds, inversion parameters, etc.). Gaussian diffusion models for a single point and for multiple points are perhaps the most popular types.

We also have meteorological models which attempt to simulate the behavior of the troposphere, including rainfall, wind direction, wind speed, air temperature, etc. These models are usually based on the input of both instrumented measurements of actual local conditions, and on historical patterns which indicate probabilities of future occurrences, based on regressions on past performance.

5.7 Allocation Models

These models allocate resources to selected points and then measure the resulting impact. An example of the GIS resources required for allocation modeling are the NETWORK allocation tools, which allow selection of a point and development of a distance/time impedance estimate in the form of equidistant contours around the point. Allocation models are typically used to allocate demand to points (as, for example, allocating students to school locations), or used to distribute resources from points to various areas requiring the resources (such as fire protection from fire stations to housing).

5.8 Optimum Route Models

One kind of these models identifies the optimum route through a spatial network such as roads. These routes can be defined between two points or among a whole group of points as in the "traveling salesman" problem. Another use of these models is identifying the optimum route for pipelines, transmission lines, etc. through a region; this requires a combination of various GIS analytical capabilities and has been recently used in combination with triangulated networks describing terrain surfaces.

5.9 Population Forecasting and Distribution Models

Population forecasting models forecast the growth of a population using cohort survival techniques and then forecast its spatial distribution based on a variety of spatial conditions and gravity related techniques.

5.10 Land Use Forecasting Models

These models are often based on population increase and industrial growth. GIS tools have emerged which can, for example, allocate new housing to various polygon-defined areas based on a variety of attributes of the polygons together with other extraneous forecasts.

5.11 Transportation Models

Transportation models allocate population movement to segments of road networks, based on other models, such as gravity models, which forecast movement between locations.

5.12 Gravity Models

Gravity models are popular for forecasting various kinds of population interactions through a network, based on the "gravitational attraction" between population masses. The chief factors considered by the model are the populations (the "masses" which attract one another) and their spatial distribution with respect to one another (assuming that distance, or some other measure of impedance, reduces the effective gravitational inter-attraction of the population masses).

5.13 Site Selection Models

These models, often based on linear programming, are designed to optimize siting of facilities. For example, one may attempt to optimize the selection of sites for industrial plants by considering such factors as each site's spatial relation to various resources, and such cost parameters as cost of ore, cost of manpower, cost of transportation, etc.

5.14 Economic Input/Output Models

Economic input/output models, involving an analysis of economic sectors and their relationships, rarely have been coupled with geoprocessing, but could be, using modern GIS tools and some of the other forms of modeling described above. This approach might be particularly powerful for analyzing the economic impacts of various environmental problems.

6. CATEGORIES OF MODELING

There are a number of different categories into which modeling efforts can be placed; the categories chosen here depend on the levels of model sophistication, intensity of use and the applications to which the modeling effort is directed. Four such levels, of ascending significance, are given here.

As indicated earlier, building relationships within a GIS database usually will precede actual modeling; relationship building will use tools which are already quite well developed and embedded in modern GIS systems. These are the tools for polygon overlay, point-in-polygon, buffer creation, distance measurement, search, etc.

6.1 Static, Deterministic Modeling

This basic level involves the simplest kinds of modeling. Reclassification, interpolation of spatial phenomena, and arithmetic and logical manipulation of data, all fit in this category. Most of these are models which are already rather well refined, well accepted and which have been in use for some time.

6.2 Simulation Modeling

These models attempt to mimic or simulate aspects of the behavior of a real system. They come in a wide variety of types, are usually dynamic models, and can be designed to deal with spatial information. There are, for example, simulation models which forecast the growth of timber stands based on various environmental parameters and on such forest inventory data as the type, mix and characteristics of stand vegetation. A simulation model of a lake may be based on compartmental modeling of the water in the lake, and may incorporate simultaneous non-linear differential equations; the model of a river or an estuary may include partial differential equations. Properly designed, such models could be used to forecast eutrophication, pollution levels, etc. This category of models has been well developed and widely used and accepted in non-spatial form and is beginning to be developed for spatial purposes. The further development of this category of models likely to be the next stage in the development of the GIMS.

6.3 Modeling the Interactions Between Systems

The next category of modeling deals with the interactions between individual systems. For example, the separate simulation models of forests, lakes and rivers, described above, might all be included within an overall model (a hierarchical model) which simulates the natural systems of a whole region. This model might be combined, in turn, with an economic input/output model of the region. This category of modeling is particularly appropriate in analyzing the complexities of many of the problems the world faces today. Unfortunately, models of this kind are just beginning to make use of spatial data, and, by their very nature, it is very difficult to validate such models. To create and validate such models will be a major task in the development of GIMS technology. Despite the obviously enormous difficulties of such an effort, such complex models may provide the least expensive means to understanding the spatial aspects of the real systems of most interest to us, and thus to predicting and ultimately controlling their behavior.

6.4 Modeling in a Decision Analysis Environment

Beyond even the creation and validation of models of interacting natural and man-made systems is the still greater challenge of crafting an environment in which such models will be used for decision analysis; this is an area in which there is not much experience. Before this can occur, the credibility of such models will have to be established and demonstrated in actual use. Effective, probably graphic, ways to present the models to decision makers will have to be devised. In order to be realistic about future uncertainties and portray the range of possibilities for decision makers, probabilistic as well as deterministic models will have to be used. This means that statistical analysis tools will have to be better integrated with our GIS systems so that such modeling can be supported more effectively. Nevertheless, acceptance will be difficult to obtain. In gaining such acceptance it must be emphasized that modeling is a tool to support decision makers, not replace them. Human beings will have to continue to use experience and intuitive judgement to handle even the most complicated situations; these Geographic Information Modeling Systems merely provide additional sources of information for the decision makers to consider. But the GIMS will become important enough that decision makers will have to become informed about what this technology can do for them.

7. FUTURE PROSPECTS FOR GIMS TECHNOLOGY

We are now seeing the emergence of more and more GIS tools that are useful and which can work even with the large data bases already in existence. Moreover, the cost of this GIS technology is falling fast. This will lead to more GIS users experimenting with modeling. Given the right environment, models will improve. Given these trends, how far modeling will penetrate into the way we manage actual operations, will depend largely on our success in perfecting the models and then providing them with user interfaces acceptable to analysts and decision makers in various disciplines. GIS's will become the linkage between this modeling environment and various operational functions.

7.1 The Need for Sound Geographic Databases

It is essential to recognize that modeling cannot improve if we do not have sound databases. Today's geographic databases have often been put together in a haphazard way. They usually lack continuity over time. They are not standardized and not well integrated; as a result, there are many inconsistencies. These flaws greatly hamper the process of calibrating and validating models. We'll need to build statistically sound, standardized, well integrated data bases, over periods of years, to support the development of GIMS technology and in order to make estimates of future events. As more GIS's are used in the day-to-day operations of organizations, and as more databases are developed for the purpose of carrying out normal operational functions, these aims may be attained more readily.

7.2 The Present Climate for Modeling

At present, based on what we see among ESRI's own users, government and private sector users seem to be concentrating on those GIS applications which offer benefits to their organizations either in reduced costs, or better performance, or both. They are collecting data, building databases, building relationships between the data in their databases, and producing practical, operationally useful graphic and text products. When they do modeling it tends to be of the static, deterministic sort. Academic and scientific users seem more interested in modeling, and modeling of the more complex sorts. (Possibly they have a clearer mandate to experiment with modeling than do the other two groups of users.) To assist such experimentation among our users we at ESRI have developed and are continuing to develop new means of sharing models and information about modeling: annual users' conferences, newsletters, ARC Macro language, new modeling tools, electronic networking, etc. We're also asking our users who are interested in modeling to tell us what additional tools and help they need from us. We believe that similar steps, taken by everyone in the GIS field, are essential to the GIMS development process.

TOWARDS AN ENUMERATION AND CLASSIFICATION OF GIS FUNCTIONS

Michael F. Goodchild

University of Western Ontario

London, Ontario, Canada N6A 5C2

ABSTRACT

The data structure used in a geographic information system can be seen as a model of spatial data. Primitive GIS data structures represent features and their attributes; more advanced data structures include explicit representation of the complex spatial relationships which may exist among objects. The analytic functions of a GIS can be seen as operations on the underlying data model. This raises the question addressed in this paper: is it possible to enumerate the set of possible GIS functions in a rigorous manner, and to organize them within some consistent conceptual framework? The paper proposes such a framework based on a relational model. A comprehensive set of GIS functions results, which can be compared to the functionalities of existing vendor products. It is also possible to describe the minimum requirements of a data base which would be capable of supporting the full range of functions.

1. INTRODUCTION

The ability of a Geographic Information System to analyze spatial data is frequently seen as a key element in its definition, and has often been used as the characteristic which distinguishes the GIS from systems whose primary objective is map production. It is widely accepted that systems for analysis require more completely developed data structures than those intended for map production; hence the distinction between "topological" and "cartographic" data structures. On the other hand there is very little agreement on the nature of such analysis; it is not clear what degree of analytic functionality is necessary to identify a system as a GIS, or what constitutes the totality of possible forms of analysis. The Canada Geographic Information System was clearly designed with two specific capabilities in mind - measurement of area and topological overlay of polygons - while more recent systems have begun to include more sophisticated forms of spatial analysis. In the past few years we have seen the first instances of location-allocation algorithms executed within a GIS as components of a spatial search application, and demand is clearly growing for more capabilities in other areas of spatial analysis.

Spatial analysis can be defined broadly as that set of analytical methods which requires access both to the attributes of the objects under study and to their locational information. It appears to be more extensive than statistical analysis, which we define for the purposes of this paper as requiring only attribute information, but the literature of spatial analysis provides no readily available principles of organization of the field, or any clear view of its extent. It seems clear that the GIS field will have to develop such principles itself if it requires them as a basis for designing appropriate functionality in spatial analysis.

Besides allowing more comprehensive GIS design, a discussion of functionality in spatial analysis has a more immediate purpose. Goodchild and Rizzo (1) have discussed benchmarking and workload estimation for GIS acquisitions using a model which decomposes workload into individual GIS operations, and Goodchild (2) has described an application of the model. The list of operations used in this work was based on familiarity with the user interfaces of current vendor products, rather than on any comprehensive view of spatial analysis. A more general list based on clearly defined principles would be more internally consistent and less dependent on currently offered capabilities.

The present paper therefore aims to provide a comprehensive framework for the organization of spatial analysis techniques, in the context of GIS. It is hoped that this will serve two immediate purposes: first, to serve as a framework for substantial expansion of spatial analysis capabilities, and second to provide the basis for organizing a generic list of GIS functions.

2. GIS FUNCTIONALITY

Much of the workload of a production GIS is concerned with input and output of data; a large proportion of the costs of operation are input-related, at least in the first few years of operation. Moreover these costs have proven to be relatively fixed, and there is little prospect of dramatic reductions in costs in the immediate future. However the focus of the present paper is on analysis, and it is analytic capabilities which fundamentally distinguish a GIS. So although input and output functions are overwhelmingly important cost components in many systems, the present discussion will focus exclusively on manipulation and analysis.

The demands placed on a GIS appear to fall into two modes, although the distinction between them is often blurred. We define the product mode as a series of GIS operations designed to generate an information product, usually hard copy, in the form of a map, table or list, from which the user will later extract selected information. The product is usually more comprehensive than the uses to which it will be put; it might, for example, contain information for an entire map sheet, of which the user will later examine only a small area. There is usually some time period over which the product remains useful, allowing the generation of products to be scheduled on a regular basis independent of the actual pattern of use. So for example a forest resource manager might request quarterly updates of the forest resource map on which most of his or her decisions will be based, but would likely not require direct interaction with the system in the periods between arrival of new quarterly updates.

The query mode assumes that the user is able to formulate a specific question, for which the answer should be provided immediately, or within some reasonable response time, and is willing either to interact directly with the system, or to give the request to an interactive operator. The question will usually be designed to provide no more than the information actually needed. Thus two dimensions distinguish product from query modes:

- immediacy, in that the query mode requires interaction with the data base whereas the product mode assumes that queries are directed to some previously extracted hard copy, and
- domain, in that the query mode assumes that the limits of the required information are known, whereas the product mode assumes only that the product will be sufficiently broadly defined to answer all likely queries.

We have already argued that spatial analysis is more general than statistical analysis, because it requires access not only to attributes but also to associated locational information. Well-known software for statistical analysis offers both query and product modes. Packages such as SAS and SPSS have developed from a product orientation, partly, one suspects, because both extend back to the days of batch processing, and partly because the complexity and volume of output of many forms of analysis requires it. It would be impractical, for example, to conduct a factor analysis in query mode. On the other hand limited statistical analysis, in the form of totals, means or bivariate displays, are available in query mode in many packages, including the more recent versions of SAS and SPSS.

It seems that the most important factor dictating the choice of mode is the complexity of the analysis. Query seems more suitable for simple analyses; more complex ones are likely to require more voluminous output, to be more difficult to specify through the user interface, to involve substantial and perhaps unacceptable response times, and perhaps to require the use of more than one package. All of these factors seem to argue for a product mode of operation for spatial analysis using a GIS. It follows that demands for more sophisticated spatial analysis are likely to be satisfied by product-oriented systems, while query mode will continue to be associated with rather limited analytic functionality. For this reason the remainder of the paper adopts a product orientation. At the present time it seems that both product- and query-oriented systems are available from vendors; the relative importance of the two modes in the vendor's perception of the marketplace dictates both the breadth of functionality and those design characteristics associated with speed of response.

3. DATA MODEL

In order to build an organizing framework for spatial analytic methods it is necessary first to establish an underlying data model. This may or may not correspond to the structures actually employed in storing spatial data in the GIS; however it will be assumed that it represents a possible view of the data from the spatial-analytic user's perspective.

The primitive elements of spatial data, representing map features, are referred to as *objects*. We assume that the data model includes point, line and area primitives. In addition the model recognizes the raster form, an ordered regular tesselation with location implied by position in the scan order. A single map coverage may have several *classes* of objects of each type; churches and windmills may both be represented by point object classes.

All object classes may have unlimited numbers of attributes. Each attribute for a given class must be homogeneous, meaning that the value of the attribute for each object in the class must be drawn from the same set of permissible values. It may be useful to classify attributes into interval, ratio, ordinal or nominal scales; we assume that the conventional methods for describing and manipulating statistical data are available for the attribute table of each object class, for example through SPSS or SAS, or to a more limited extent through data base management systems such as ORACLE or INFO.

Associated with each type of object are a number of *reserved fields*, which have fixed meanings. These may be geometrical; for example for an area data type there might be reserved fields for measures of area and perimeter length. Other reserved fields have topological significance: each edge or arc of a line data type in a network might carry pointers to the nodes or point objects at each end, and to the polygons or area objects on each side; polygon objects carry pointers which allow retrieval of the line objects from which they are constructed. We assume that the data base codes topology at least to the extent of allowing retrieval of adjacencies between pairs of both area and line objects.

Note that the structure of fixed attribute tables implies that the data base can code 1:1 and n:1 relationships with ease, but not

1:n. For example it is easy to carry pointers from each edge to adjacent nodes in a network, but not from nodes to adjacent edges. However the data model being described is merely the user's view; the actual data structures used internally to represent topological relationships may well include 1:n mappings.

The model must also include the object-pair as a recognized data type, because of its fundamental importance in spatial analysis. It must be possible to take two classes of objects, which may or may not be of the same type, and create a new "object" of each unique pair. Object-pairs may combine any types, although pairs which include raster types seem unlikely to have much practical application. The reserved fields of object-pairs include pointers to the objects which created each pair, together with distance between the objects. Other attributes might include flows of goods, numbers of migrants or trips, or details of transportation links. Although object-pairs have associated distances, these must be evaluated using a metric as the object-pair has no physical existence, unlike a line object.

Higher levels of combination are possible, beginning with the object-triple and including pairs of object-pairs, but there appear to be few applications in spatial analysis; the triangular inequality seems to be the only well-known property of objects (or object-pairs) taken three at a time.

The object/attribute model provides a simple view of spatial data, and appears to be relatively comprehensive when augmented with the concept of object-pair. It allows conventional statistical packages to be seen as processors of single, rectangular attribute tables; the more complex hierarchical, network and relationship models are supported by numerous data base management systems but seem to have had insignificant influence on statistical analysis.

However there are examples of data structures which occur in certain areas of spatial analysis and which are not included in any obvious way in the model. One is the path, which results, for example, from the solution of a shortest path problem, and can be viewed as an ordered set of network links or edges. A path might be represented in several ways in the model: as a binary attribute of each link, indicating whether the link was or was not included in the path; as an attribute of each link interpreted as a pointer to next link; or as a binary attribute of an object-pair composed of links paired with links, indicating whether each pair of links is adjacent in the path. The first of the three alternatives would require a search to recover the correct order of links, whereas order is coded directly in the other two. The tree structure is another example; it results when the shortest path problem is solved from one origin to all possible destination nodes in a network; it is easy to represent paths from destination to origin as link or node attributes interpreted as pointers, but the reverse requires 1:n capabilities.

4. GIS OPERATIONS

Since spatial analysis is seen in this discussion as including statistical analysis, the focus of this section is on the additional operations which are necessary to support the former; the ability to carry out statistical operations is assumed to exist in the form of standard statistical package capabilities.

The proposed system of organization of GIS operations has two levels. At the highest, operations are classified according to whether they:

a) analyze the attributes of a single class of objects (statistical analysis);

b) analyze one class of objects using both locational and attribute information;

c) analyze the attributes of object-pairs;

d) analyze more than one class of objects;

e) create new object-pairs for one or two existing classes of objects; or

f) create a new class of objects from one or more existing classes of objects.

In cases of new object creation it is assumed that reserved fields will be filled appropriately; in all cases operations are assumed to generate additional attributes for the new or existing objects as appropriate.

The lower of the two levels of organization concerns the types of objects being processed in each class. Thus the creation of an area object type from a point type is distinct from creation of a line type from a point type, but both are examples of group (a) operations above.

The definitions derive from operations on the data model, rather than from the cartographic meaning of those operations. For example, the pairing of points in a class of objects representing churches might generate an object-pair in a group (e) operation, a reserved field recording the distance between each pair of churches, or might generate a line object class of straight line segments between each pair in a group (f) operation, with a reserved field of line length. The effects on the data model are quite different, although the operations are conceptually similar.

The next section of the discussion analyzes a number of example operations in terms of this framework, in increasing order of complexity. This is followed by a more extensive discussion of one particularly important operation, polygon overlay.

5. EXAMPLE OPERATIONS

5.1 Boolean Selection

The selection of a subset of objects from a class based on their attributes requires no access to locational information; it falls into class (a) above.

5.2 Nearest Neighbour Analysis

Various nearest neighbour statistics can be computed for a point object class by processing their locational information in a group (b) operation. Alternatively the same statistics can be computed from the distance reserved field of point-point object-pairs in a group (c) operation.

5.3 Spatial Autocorrelation

Calculation of spatial autocorrelation for a class of areas, using either the Geary or Moran coefficients (3), would be structured as a group (d) operation if weights are to be based on polygon adjacencies. The operation requires access to the area attribute for which spatial autocorrelation is to be computed, together with the adjacent polygons for each arc in the line object class representing polygon boundaries; two object types, areas and lines, are therefore required, together with the necessary topological links between them in the form of reserved fields.

5.4 Thiessen Polygons

Many GISs include the capability of creating Thiessen or Voronoi polygons from point objects, and the literature describes various generalizations (4), as well as numerous applications to spatial data handling. In vector mode the operation falls into group (f); the new area objects created would have standard reserved fields, and would likely include a pointer to the generating point as attribute.

5.5 Spatial Interaction Modelling

Openshaw (5) has commented on the growing gap between the level of theoretical development in many areas of spatial analysis, and the corresponding level of operationalization and practical application. One of the difficulties contributing to this appears to be the inability of standard statistical packages to handle the complex data models of many forms of spatial analysis. The calibration of a spatial interaction model is a case in point; it requires attributes of point objects representing destinations, area objects representing origins and point-area object-pairs representing trips. The need to handle multiple types of objects places this operation in group (d). The results would be returned as summary statistics or as new attributes of the existing objects; for example predicted numbers of trips would be a new attribute of the point-area object-pair.

5.6 Transportation Network Analysis

The large number of techniques for analysis of transportation networks provides a good example of the rapidity with which spatial analysis capabilities can expand when combined with a spatial data base. The origins and destinations of the system can be represented as point objects, and the connections between them either as lines with appropriate pointers or as point-point object-pairs, depending on whether flow is constrained to follow the links of a finite network. Goodchild and Lee (6) have recently programmed the following four classes of transportation algorithms as group (d) operations for the Electronic Atlas of Canada:

- shortest path analysis;
- optimum tour routing;
- location-allocation;
- transportation and transshipment problems.

Each algorithm receives node and link attributes from the data base and returns the results of the analysis in the same form, as new link or node attributes, allowing existing capabilities of the system to provide input and output support and other functionality.

5.7 Shortest Path Analysis

The finding of the shortest path through a network from an origin node to a destination falls into group (d), as it requires access to both point and line objects, representing nodes and links respectively. Results can be returned in several forms; the total distance from origin could be returned as a node attribute, and the shortest path could be returned either as pointers from node to next node, or from node to next link (link to next node would likely be available as a reserved field). A more developed version of the shortest path problem allows the addition of impedances in turning from one link to another at each node; this "turntable" might represent delays due to having to slow a vehicle, or might be used to prohibit illegal turns. The turntable can be represented as an attribute of a line-line object-pair.

6. POLYGON OVERLAY

The case of polygon overlay seems sufficiently rich and important to merit special attention. Superficially this is a group (f) operation, involving the creation of a new class of area objects from two (or more) existing classes. However the results of the operation can be returned in a number of ways:

- new area objects with reserved fields, and either the concatenated attributes of the generating areas, or pointers back to them;
- new point objects at the intersections of the existing area objects, each with four pointers to the areas whose common boundaries intersected;
- new line objects, being the old line objects broken at intersections and having either the attributes of the old line objects or pointers back to them;
- new attributes for the existing areas pointing to the new areas formed from them;
- a binary attribute of a new area-area object pair indicating whether or not the pair of areas intersect.

It is clear that not all of these will be relevant in any application, suggesting that the system designer might provide a list of appropriate options. However the set of information required to some extent dictates the algorithm to be used. This leads to a potential conflict between the need for consistency on the one hand, which places all polygon overlay operations together, and the need for operational efficiency on the other. A case in point is the window operation; the overlay of an area object (the window) on a set of area objects. Since the user is not interested in those objects falling outside the window, the system designer is able to provide a special algorithm for windowing which is generally faster than the full polygon overlay. Consistency with the organizing framework is lost if this algorithm is invoked by a different command, but not if the system is able to identify which algorithm to use in each case.

7. CONCLUSIONS

The true potential value of Geographic Information Systems lies in their ability to analyze spatial data using the techniques of spatial analysis. Although an enormous amount of development has taken place in spatial analysis in the past two or three decades, much of that work remains largely unapplied due to the inherent difficulties of processing spatial data, with its complexity of relationships. GIS provides the key to this problem through a consistent and relatively complete model of representation, and through the economies of scale inherent in the sharing of many input, output and housekeeping functions. GIS development now seems to have reached the stage where rapid expansion of spatial analytic capabilities is possible, in such fields as spatial search, spatial decision making and spatial interaction modelling. So for example the potential exists to apply many of the theoretical developments of the 1970s in spatial interaction modelling to retail site selection.

Since the set of techniques in spatial analysis is so large, a modular approach to development is clearly necessary, a pattern seen already in the TIN and NETWORK additions to the ARC/INFO data base. To be fully effective, modular development requires that the base be complete, and it appears from the analysis presented in this paper that that point has not yet been reached by any available system. When it is, we can expect very rapid expansion of capabilities, since there will be a major incentive for any theoretician or software developer working in the field of spatial analysis to provide implementations consistent with the data model; this pattern is evident in the history of SAS and more generally in third-party software development for the IBM-PC.

References

1. *GOODCHILD, M.F. & B.R. RIZZO*, 1987: Performance Evaluation and Workload Estimation for Geographic Information Systems. International Journal of Geographical Information Systems, 1: 67-76.

2. *GOODCHILD, M.F.*, 1987: Application of a GIS Benchmarking and Workload Estimation Model. Papers and Proceedings of Applied Geography Conferences, 10.

3. *CLIFF, A.D. & J.K. ORD*, 1981: Spatial Processes: Models and Applications. Pion, London.

4. *BOOTS, B.N.*, 1987: Modifying Thiessen Polygons. The Canadian Geographer 31: 160-8.

5. *OPENSHAW, S.*, 1987: Guest Editorial: An Automated Geographical Analysis System. Environment and Planning A 19: 431-6.

6. *GOODCHILD, M.F. & J. LEE*, 1987: Network Algorithms for the Electronic Atlas. Energy, Mines and Resources Canada, Ottawa: Final Report of Contract 2ER.23242-6-2385.

BLANK PAGE

APPLICATION OF THE OBJECT-ORIENTED PARADIGM TO PROBLEMS IN GEOGRAPHIC INFORMATION SYSTEMS

Kenneth J. Dueker
Daniel Kjerne

Center for Urban Studies
Portland State University
Portland, Oregon, USA

ABSTRACT

The object-oriented paradigm allows problems to be recast in terms closer to those used by experts in the problem domain, with less concern for details of hardware and software implementation. In contrast to database management systems based on procedural languages, entity behavior can be modeled within the database rather than outside it.

Progress on geographic information systems using an object-oriented paradigm has concentrated on issues of system design, with relatively little discussion of applications. This paper identifies three researchable problem areas that could be fruitfully investigated using an object-oriented approach.

The first area, modeling of locational behavior of cadastral objects, is underway and shows promise; the paper details how an object-oriented approach has facilitated progress. The second, generalization from large-scale models (databases) to small-scale displays (maps) is in a nascent stage at present, awaiting a large effort in development of knowledge of rules for object generalization; suggestions are offered as to how an object-oriented approach may prove helpful. The third area, development of GIS expert systems, is also not well developed; the paper discusses how it may benefit by linking object-oriented GISs to already-developed expert system shells written in an object-oriented language.

1. INTRODUCTION

It has been said that learning how to program in an object-oriented language is more difficult than learning how to program in a procedural language – particularly (ironically enough) for those already proficient in procedural languages. Actually, programming is programming, and the skills learned in writing Fortran are easily transferred to similar operations in Smalltalk. The real problem is that object-oriented programming uses a different metaphor than other styles of programming, one that appears to people new to any kind of computer language to be a more natural one for dealing with models of systems. But programmers experienced in other languages have to unlearn a few habits before they feel comfortable at what's going on. Below is a concise review of the principles of object-oriented languages:

> In an object-oriented environment, …anything which is to be represented within the system is an object. An object consists of a private memory with a public interface. Each object within the system is identified by a unique system-defined object identifier [that] does not change even though properties of the object may change.
>
> An object's private memory cannot be directly accessed or manipulated by other objects. Any operation upon an object's private memory can only be effected by sending an appropriate request to the object. Since each object has a well-defined interface, and sole control over its own private memory, an object actually functions as an abstract data type.
>
> Many objects within the system exhibit similar behavior. These objects are grouped together in classes. Each object within a group is said to be an instance of the class of that group. Classes are arranged in a superclass/subclass hierarchy. The behavior defined within each class is inherited by its subclasses.
>
> Each class defines the behavior of its instances. However, the instances are not identical. Instances are distinguished by their unique identifiers and the contents of their private memories. The set of classes is not fixed. Users can create new classes as well as new instances of current classes. (1)

Designing an object-oriented database management system (DBMS), just as in designing a DBMS in any language, requires resolution of issues of persistence of entities in the database, concurrent actions on the database, system resilience to abnormal termination of processes, and so forth. The difference lies in how the DBMS lends itself to modeling the data world of interest to the user:

> Today's database systems are mostly based upon one of the now classical data models (hierarchical, network, or relational) or one of their derivations. As these models are all tailored to account for the representation of rather simple entities only, the semantic gap tends to become large when complex entities need to be dealt with. This is at least partially due to the fact that in these cases *one* conceptual miniworld entity has to be represented by a *number* of database objects (e.g. records, tuples, …)
>
> This leads us to a first-cut informal definition of an object-oriented database system: it is based on a data model *that allows* [the designer] *to represent one miniworld entity* (whatever its complexity and structure) by *exactly one object* (in terms of the data model concepts) *of the database*. Thus no artificial decomposition into simpler concepts is necessary in any case (unless the database designer decides to do so). Note that as entities might be composed of subentities which are entities in their own right, an object-oriented data model also has to allow for recursively composed objects. (2)

Geographic information systems (GIS) intended for more than single-purpose use need to be capable of handling questions of consistency, error detection, updating, etc., just as any general purpose DBMS. In addition, they must be capable of answering topological, metrical, and geographical questions about the entities in the database (3). Entities possessing topological and metrical characteristics certainly strain the representational capabilities of currently available DBMSs. For instance, GISs now split up the graphic and non-graphic (attribute) data relating to an entity. A thorough-going object model would integrate both into a single data type.

Doubtless it is because an object-oriented geographic information system has yet to become available outside the development laboratory that published work concentrates on issues of system design (4, 5, 6). Nonetheless, there is some agreement that this new way of doing things promises to be effective and powerful. Presumably, once the availability situation changes, we shall be deluged with solutions to a number of heretofore difficult or intractable problems in capturing, manipulating, and presenting spatial data.

In the absence of operational object-oriented geographic information systems, it is only possible to speculate on what conceptual entities would be defined as objects in a database. That, at any rate, is the plan for this paper, with the aims of arousing some interest in making use of this approach and bringing out suggestions for strategies.

We will examine three problem areas: modeling of locational relationships; map generalization; and expert systems in GIS.

2. MODELING LOCATIONAL RELATIONSHIPS

Whether we use a planetable, theodolite, or photogrammetric plotter to build a base map, or trace boundaries from an atlas to compile a thematic map, the fundamental process of mapping locations is the same: decide upon the locations of a class of **control** entities and measure the relative displacements of another class of **mapped** entities. The act of measurement for any mapped entity entails

- performing **one** of a set of **procedures,**
- finding the location of **one or more entities** that are (for that mapped entity) control entities, and
- generating **zero or more values** associated with the particular mapped entity.

To state it another way: every thing on a map is where it is for some determinable reason. This can be seen even for things appearing on cadastral (property) maps: the reasons for things being where they are can be legalistic in nature as well as having a basis in measurement operations conducted in the real world; nonetheless, when a cadastral cartographer places an object in a specific location, it should be possible to determine the chain of reasoning that led to that choice, and to store those reasons in the database.

Perhaps this could be accomplished using a conventional database. In a relational data model, one might have a "locational tuple" containing the entity's ID, the control entity or entities' ID, the name of the locational procedure, and any values. To determine the current location would entail getting and performing the procedure referenced for that entity, plugging in the control and values. But the procedure could not reside in the database itself (7).

It seems that the concept of an object containing executable methods as well as data is the simplest way out of this. In work proceeding at present (8), object classes for cadastral maps have been defined that, when requested for their location, derive it by requesting the location of their control entities, performing the procedure stored for their object class, and applying the values contained in their internal variables.

The implementation of cadastral mapping requires a broader conceptualization of point-line-area relationships. In the "real world," areas of various types (e.g., statistical, political jurisdictions, watersheds, land parcels) may have various types of boundary lines (e.g., natural feature edges or right-of-way, road center, survey, or parcel boundary lines). In turn, these lines may be bounded by various types of points (monuments, property corners, intersections). A locational change to one of these elements will impact the location of bounding elements. The chief advantage of object-oriented modeling of these relationships will be in an improved correspondence with our concepts of the real world.

3. GENERALIZATION

The first problem area looked at capturing more knowledge about where an object is. This second area looks at capturing more knowledge about how to present an object graphically at various scales.

The problem here has to do with the display of a large-volume database at various scales, paying attention to the characteristics of different lines and symbols.

With regard to lines, a procedural approach lends itself to two possible strategies. The first stores line data – for instance for a whole coastline – as a unit, and relies on a sophisticated algorithm to treat different portions differently for generalization. It could test portions of a line to find the point density or degree of curvature in that portion, then apply the appropriate procedure to decide which points not to display at small scales.

Alternatively, one might define a large number of data types – fjord, barrier island, delta – and a set of display procedures, one for each data type at each scale. Display would then entail calling the appropriate procedure and naming the entities to be acted upon by it.

The object-oriented approach is actually similar to the latter strategy, with the difference that the methods (procedures) are bound to the data types, so that instead of having to know what data type you are dealing with ahead of time, you can request objects to display themselves and leave to them the details of how they do it. Thus, each object class – fjords, braided streams, towns smaller than 5,000 – would "know" what it should look like at a given scale.

The work to be done here involves determining what methods are appropriate for different classes of mapped objects at different scales and deciding what class particular objects fall into for a particular map. This gets us into the area of knowledge engineering: how do cartographers decide to do things the way they do?

4. EXPERT SYSTEMS

In the previous two sections, we looked at problems of capturing more knowledge about location and display. If we "go up one level" to investigate the rules used to decide what the location for a given object should be, or how to display a given object at a given scale, we enter the realm of expert system design.

Application of expert system technology to geographic information systems awaits results in a number of research areas, identified by Robinson, Frank, and Blaze (9) as:

> [1] improved facilities for editing, viewing, and organizing knowledge in an ES [expert system] and extension of ES shells to work with very large collections of spatial data and larger collections of rules than presently feasible;
> [2] it must become possible to incorporate simple procedural operations, including numerical computations, into existing symbolic processing systems;
> [3] several of the concept areas in geography must be explored and become more formalized….
> [4] as formalization of geographic concepts … progresses, we will need to formalize the different types of uncertainty in geographic knowledge bases …

Both problem areas mentioned above -- location and display -- would seem to be responsive to an antecedent-driven or forward-chaining strategy in which, as described by Michaelson, Mitchie, and Boulanger, "…you enter the information into the database, and the system then applies the rules to reason from the data to the conclusions." (10) These authors note that object-oriented programming is well suited to a forward-chaining strategy:

> Some forward-chaining systems try to control the search for rules in the recognize cycle by grouping rules into packets. These rule groupings are appealing conceptual structures, since they group rules according to the subtopic with which they deal. Object-oriented programming can also be used to organize collections of rules. In object-oriented programming, we give objects behavior, and thus we can distribute the control of rules into rule, rule-packet, and domain objects. This approach …also allows multiple instantiations of the same set of rules to solve subproblems of the same type within one session.

5. CONCLUSION

In earlier work, we have said that it would be easier to build upon an object-oriented GIS, presuming that such a system would have a number of predefined classes incorporating display methods. These could then be used as the basis for subclasses with new methods for, *e.g.*, computing locations or displaying at different scales. It now appears that claims of "object orientation" have to be examined carefully, as GISs promoted by vendors may not be capable of defining new subclasses or performing other operations basic to object oriented design (11).

On the other hand, building a mapping application from an object-oriented database management system offers the prospect of defining object classes with basic display methods. For systems based on highly graphic interfaces (such as descendants of Smalltalk) this is less daunting than might be for procedural languages, and the definitions should be reusable.

In fact, for initial development of object class definitions useful to geographic data analysis, database-related capabilities (such as object persistence) are irrelevant. The classes could be defined in the DBMS's native language and moved to it when a specific project was to be implemented.

It has been noted that the object-oriented approach is suited to prototyping, but that applications are, in general, slower than ones written in procedural languages. Would it make sense to use this approach only in exploratory, prototyping work, and use a faster procedural-based DBMS once the data model was defined? This approach might be viable for cases where the data model was not subject to change or extension - but that is not a general characteristic of the kinds of problems GISs model.

Speed, in any case, is relative. Foreseeable improvements in hardware and software will make what now appears slow to run reasonably quickly. Of course, things that now run reasonably quickly will, by the same token, run even faster.

The availability, in the next year or so, of data model compilers and the growing sophistication in methodology of data model design for object-oriented DBMSs (12, 13) increase the possibilities in the future for generating database software corresponding to user-specified data models. Given this trend, and the difficulty major turnkey vendors have had in integrating object-oriented approach with existing computerized mapping approaches, we may see a tendency for users to become independent of monolithic system solutions. Consultants for agencies, or agencies themselves, will likely take generic (and more hardware-independent) object-oriented database management systems and modify them to their more specific needs. This empowerment of application users to design their own environments was foreseen early on by the developers of object-oriented language (14).

REFERENCES

1. *KETABCHI, M., & WEINZ, R.,* 1987: Implementation of Persistent Multi-User Object-Oriented Systems. Proceedings: COMPCON Spring 1987. IEEE Computer Society Press: Washington, D.C.;

2. *DITTRICH, K.,* 1986: Object-oriented Database Systems: the Notion and the Issues. Proceedings: 1986 International Workshop on Object-Oriented Database Systems. IEEE Computer Society Press: Washington, D.C.;

3. *WHITE, M.* 1984: Technical Requirements and Standards for a Multipurpose Geographic Data System. The American Cartographer, 11 1: 15-26;

4. *MANOLA, F., ORENSTEIN, J., & DAYAL, U.,* 1987: Geographic Information Processing in the Probe Database System. Auto-Carto 8 Proceedings. ASPRS and ACSM: Falls Church, VA;

5. *LOCHOVSKY, F.,* 1985 (ed.): Special Issue on Object-Oriented Systems. Database Engineering, 8 4;

6. *WEINTZEN, B.,* 1986: TIGRIS: An Object-Oriented Approach to Topology. InterVue, 7 4: 14-18;

7. *SMITH, K., & ZDONK, S.,* 1987: Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems. OOPSLA '87: Object-Oriented Programming Systems, Languages and Applications Conference Proceedings. ACM Press: New York;

8. *DUEKER, K.,* 1987: A Rule-Based Approach to Cadastral Mapping Using an Object-Oriented Geographic Information System. (Project proposal submitted to the Geography and Regional Science Program, National Science Foundation);

9. *ROBINSON, V., FRANK, A., & BLAZE, M.,* 1986: Expert Systems Applied to Problems in Geographic Information Systems: Introduction, Review, and Prospects. Computers, Environment, and Urban Systems, 11 4: 161-173;

10. *MICHAELSON, R., MITCHIE, D., & BOULANGER, A.,* 1985: The Technology of Expert Systems. BYTE, 10 4: 303-312;

11. *PASCOE, G.,* 1986: Elements of Object-Oriented Programming. BYTE, 11 8: 139-144;

12. *MARYANSKI, F., BEDELL, J., HOELSCHER, S., HONG, S., McDONALD, L., PECKHAM, J., & STOCK, D.,* 1986: The Data Model Compiler: A Tool for Generating Object-Oriented Database Systems. Proceedings: 1986 International Workshop on Object-Oriented Database Systems. IEEE Computer Society Press: Washington, D.C.;

13. *LYNGBACK, P., & KENT, W.,* 1986: A Data Modeling Methodology for the Design and Implementation of Information Systems. Proceedings: 1986 International Workshop on Object-Oriented Database Systems. IEEE Computer Society Press: Washington, D.C.;

14. *GOLDBERG, A.,* 1981: Introducing the Smalltalk-80 System. BYTE, 6 8: 14-26;

BLANK PAGE

ISSUES IN THE USE OF EXPERT SYSTEMS TO MANAGE UNCERTAINTY IN GEOGRAPHIC INFORMATION SYSTEMS*

Vincent B. Robinson
(vbr%survey@uncaedu.bitnet)

Ross Miller
(miller%survey@uncaedu.bitnet)

Laurie Klesh
(klesh%survey@uncaedu.bitnet)

Department of Surveying Engineering
The University of Calgary
Calgary, Alberta T2N 1N4
CANADA

ABSTRACT

This paper outlines some of the thinking behind a project for investigating the use of expert systems for managing uncertainty in geographic information systems. Different theories of uncertain reasoning in expert systems are addressed. Strategies for dealing with uncertainty in polygon overlay processing and spatial query are discussed. Problems such as knowledge acquisition and processing overhead are mentioned.

1. INTRODUCTION

Uncertainty in Geographic Information Systems (GIS) is a complex and persistent problem with many sources (1). We report on a project to develop a framework for handling uncertainty in GIS. One of the major obstacles to the coherent management of the different types of uncertainty in GIS has been the lack of tools for explicitly representing and dealing with uncertainty. Most contemporary systems treat uncertainty in an implicit manner which results in the burden of uncertain information processing to be imposed on the human user (1) or causes the consequences of uncertainty to be absorbed by society (2).

We have organized this project into three phases. First we will investigate general models of uncertainty in expert systems which may prove pertinent

---

* This research is partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada and by the Alberta Forestry, Lands, and Wildlife Professorship in Digital Mapping and Spatial Data Management.

to GIS. The second phase consists of investigations into specifying the rules needed in the expert GIS. We will concentrate on the polygon overlay process (POP) and the spatial query processing (SPATQ) problems. The final phase is development of a prototype expert GIS. A major obstacle to be overcome in this phase will be the acquisition of knowledge that can be used in the management of uncertainty.

2. EXPERT SYSTEMS AND GEOGRAPHIC INFORMATION SYSTEMS

Expert system technology provides tools for advancing the formal representation and management of uncertainty. Such systems allow not only facts, certain and uncertain, explicit representation in the knowledgebase but also rules for the inference of additional information from the knowledgebase. The power of the expert system approach lies in the ability to develop a comprehensive and realistic system for managing the different kinds of uncertainty in GIS while incorporating heuristics that may be exploited by human users of contemporary systems. The representation of uncertain knowledge (i.e. facts and rules) is maintained independent of the inference mechanism. The knowledgebase is therefore able to represent domain-specific knowledge while the inference mechanism is domain-independent.

Robinson et al (3,4,5), and Robinson and Frank (6) have discussed at length expert systems and their potential for application in GIS. They consistently identify the formalization of geographic knowledge and management of uncertainty as two areas particularly well-suited for the application of expert systems to problems of GIS. In particular, Robinson et al (5) identified two major problems of contemporary GIS - user interface and overlay processing. Both of these problems require significant domain-specific knowledge that is difficult to represent and maintain using more traditional methodologies. This is particularly relevant when considering the management of uncertainty in a GIS environment because many issues of uncertain geographic knowledge are sensitive to a particular domain. Before discussing problems of uncertainty in overlay processing and spatial query we present some of the approaches for handling uncertainty in expert systems.

3. UNCERTAINTY IN EXPERT SYSTEMS

The ability to handle uncertainty is considered by many to be an essential characteristic of expert systems. Here we provide an overview of some of the major approaches being considered for its management in CIS. These include the use of certainty measures, approximate reasoning, evidential reasoning, and Bayesian probability theory. Traditionally, these approaches were considered as separate and competing. The arguments of Yager (7) and Shafer (8) in support of a more unified approach will be summarized. Finally, some persistent problems related to handling uncertainty in expert systems are mentioned.

3.1 Certainty Measures

An early expert system, Mycin, used an empirical model for managing uncertainty. The Mycin certainty factor model has since become quite popular even though the certainty measures have little theoretical underpinning (9).

In Mycin a certainty factor, $Mcf(h,e)$, is attached to each rule representing the change of belief in the hypothesis h given the evidence e. Mcf ranges between -1 and +1. Positive numbers represent an increase in belief. Negative numbers represent a decrease in belief.

In terms of probability the formal definition proposed by creators of Mcf is :

$$Mcf(h,e) = \begin{array}{llr} (p(h,e) - p(h)) \,/\, (1 - p(h)) & \text{if} \quad p(h,e) >= p(h) & (1a) \\ (p(h,e) - p(h)) \,/\, p(h) & \text{if} \quad p(h,e) < p(h) & (1b) \end{array}$$

where $p(h,e)$ is the conditional probability of h given evidence e and $p(h)$ is probability of h before e is known, i.e. prior probability. Thus, Mcf do not represent an absolute amount of belief in an hypothesis, rather a measure of the update of a belief.

There are two basic patterns of reasoning with uncertainty. One is deductive inference from uncertain premises. This is known as sequential combination and is used when an hypothesis e serves as evidence for another hypothesis, i.e. $h(e' \longrightarrow e \longrightarrow h)$. The equation for it is :

$$Mcf(h,e') = Mcf(h,e) \max(0, Mcf(e,e')). \qquad (2)$$

The second pattern is that of parallel combination. Here we have two pieces of evidence e' and e" that both bear on the same hypothesis h. The equation used for parallel combination is :

$$Mcf(h,e',e'') = \begin{array}{llr} c' + c'' - c'c'' & \text{if } c',c'' \geq 0 & (3a) \\ c' + c'' + c'c'' & \text{if } c',c'' < 0 & (3b) \\ (c' + c'') \,/\, (1 - \min(c', c'')) & \text{otherwise} & (3c) \end{array}$$

where $c' = Mcf(h,e')$ and $c'' = Mcf(h,e'')$. Equations (2) and (3) are propagation functions (Mcf) because their meaning to propagate belief through the inference network of hypotheses and evidences.

There is an important problem associated with the use of Mcf. Mcf's do not respect requirements for probabilistic interpretation. More to the point, their definition dictates a noncommutative parallel propagation function. Thus, in the Mycin model the way of updating belief depends on the order in which the evidence is combined.

3.2 Evidential Reasoning

Evidential reasoning (ER) is based on the Dempster-Shafer theory of belief functions. Belief functions are provided by knowledge sources (KS) and represent their view of the state of the world. A unit of belief, relative likelihood, is distributed over a set of propositions describing possible states of the world. The set of possible propositions over which a KS can distribute its unit of belief is the set of all subsets derived from the frame of discernment. The frame of discernment is a set of mutually exclusive, exhaustive, primitive propositions. Belief functions, or evidential mass functions (Emf), may distribute their unit of belief over nonexclusive propositions. A KS may indicate ignorance or imprecision by providing a mass function that distributes belief over a disjunction of propositions without providing a detailed distribution of belief to elements of the disjunction.

Using evidential reasoning offers several advantages. One advantage relates to a KS's ability to state precisely what it knows about a situation without being forced to use artificial precision by the mathematical imperatives of classical probability theory. This theory does not require the knowledge or use of prior probability distributions. Thus, it is relatively straightforward to construct the underlying knowledge structures.

Assuming we have pieces of evidence, the inference mechanism used in ER has been summarized as:

(1) represent each piece of information as a belief structure;
(2) use Dempster's rule to combine these pieces of information;
(3) use the projection to obtain a belief structure over any frame of concern (7).

3.3 Approximate Reasoning

The theory of approximate reasoning (AR) is strongly rooted in the theory of fuzzy subsets. This theory manipulates information in a manner that generalizes predicate calculus in a number of ways. Among the most prominent is that it considers predicates as sets rather than as individual objects - in a manner similar to that used in the relational algebra of database theory (10). AR extends predicate calculus by using the fuzzy subset instead of the crisp subset. Thus, introduction of fuzzy subsets brings a flavor of multivalued logics, while predicate calculus and relational algebra deal only with binary logic. Furthermore, the introduction of fuzzy subsets allows for a natural representation of the types of semantic imprecision that play an important role in human reasoning.

Assuming that we have a collection of propositions that are translatable under this theory, the fundamental idea of AR's reasoning mechanism can be summarized as :

(1) represent each proposition as a possibility distribution;
(2) conjunct the collection of propositions (using the extension principle);
(3) then use the projection principle to obtain the possibility distribution of any variable of concern (Yager, 1986).

3.4 Unification of Approaches

Other researchers have demonstrated how some of the above approaches can be unified. Garvey et al (11) briefly describe an inference technique that appears to provide the capability for probabilistic reasoning when appropriate underlying models are available, evidential reasoning when incomplete descriptions must be used, and logical (Boolean) reasoning when truth values of propositions are only true and false. Yager (7) has shown that in the face of nonspecific and fuzzy uncertainty AR and ER provide equivalent frameworks for reasoning. Nonspecific uncertainty is where the value of a variable is narrowed down to an interval, while fuzzy uncertainty concerns concepts that have imprecise, or gray, boundaries.

Baldwin (12) describes a support logic programming system that may represent a tangible unification of AR and ER. It uses a theory of support pairs to model various forms of uncertainty. Uncertainty is associated with facts and rules by way of a pair of supports. Support logic programming uses ideas from fuzzy set theory and evidence theory. An important aspect of this system is the manner in which facts not in the knowledgebase are treated. Facts not in the knowledgebase are uncertain. This is in contrast to most other systems which are based on the closed world assumption that facts not in the knowledgebase are false.

3.5 Acquisition of Uncertainty Representations

One of most persistent problems associated with the explicit modeling and and management of uncertainty in expert systems has been the question of

"Where do certainty numbers come from ?"

This question can drive AI researchers to despair. Tversky's (13) work shows how biased even experts can be when assessing uncertainty. This work seems to have been tacitly ignored by those who persist in building systems where experts actually input their estimates of uncertainty/certainty directly into the knowledgebase of an expert system.

Kosko (14) suggests that his theory of fuzzy knowledge combination allows for several weighted experts to determine uncertainty weights. In other words, different subsets of experts with different credibility weights can be used for different rules. It is suggested that a partial answer to the

BEST COPY AVAILABLE

problem of determining certainty numbers is by using multiple weighted experts concurring on an if-then relationship. This is not a new suggestion. Gaglio et al (15) discussed various aspects of constructing multiperson concepts for expert systems. Robinson and Wong (16) reported on an effort to construct multiperson spatial concepts of spatial relations.

4. UNCERTAIN SPATIAL RELATIONS

GIS are constructed for the purpose of answering questions about a region of the earth's surface. There are two fundamental operations which allow such questions to be answered - polygon overlay and spatial query. These two areas have been specifically identified as areas where progress must be made in order to make GIS more functional in solving real-world problems (5). Our perspective on the problems of managing uncertainty in GIS is based on the characterization of overlay and query as problems in uncertain spatial set relations.

4.1 Polygon Overlay Processing (POP)

As suggested by (17), the POP problem in GIS can be broken into two fundamental components. One is the determination of the geometric intersection of two geographic entities. The other is the assignment of attributes to the resulting composite entities. What makes the POP problem so attractive for the application of expert systems is the amount of domain-specific processing that often must be done. The fundamental operations of POP are rather simple, but as much as 80% of the code for some overlay systems is devoted to handling special cases, input, output, and book-keeping (18). Thus, there is a fair amount of knowledge pertaining to the geometric aspects of POP.

It is often convenient to separate the geometric portion of the POP problem from the semantic overlay problem. However, separating these operations does not mean they are independent. However, a problem long recognized but ignored is that the nature of the geographic entity can often influence the amount of certainty we can attach to the boundaries of that entity. This interaction effect is discussed in more detail in (1).

Modeling the management of multi-layer polygonal uncertainty affects a wide variety of data analysis operations in GIS. Among those operations affected are point-in-polygon, transects, profiles, and calculation of distance, area, and direction. There are implications affecting the responses to spatial queries.

There are a number of different possible situations which give rise to differing rule sets. However, we characterize two major approaches to managing uncertainty in overlay processing. One, advocated by (19) is based on a clear hierarchy of certainty. We also recognize that there may not be a clear hierarchy of evidence. In the latter case, a unified approach such as support logic programming applied to POP may be appropriate.

The spurious polygon problem (SPP) continues to seriously affect the utility of GIS that rely on POP for integrating data layers. We have identified four major approaches used to resolve SPP. They are:

I.model - ignore very small spurious polygons.

HW.model - use hierarchical, weighting management scheme.

AF.model - use arbitrary function.

HE.model - interactive editing where a human operator works through polygons 'cleaning them up'

The AF.model is essentially procedural while HE.model is a declarative solution to SPP. In HE.model the operator must take into consideration, among other things (from 19):

a. significance of the relation to the project,
b. geometric consistency,
c. cartographic design standards,
d. agency policy,
e. certainty related to
    1. attributes of source polygons
    2. nature of polygon boundaries.

HW.model is an intermediate approach which is more closely declarative than procedural and represents a tentative first step in formalizing a knowledgebase approach to solving SPP. It is interesting to point out that the hierarchical weighting solution is quite similar to taking the union of two fuzzy subsets in fuzzy logic. Thus, it seems that a better understanding of the formal theories of approximate reasoning could be developed to provide the formal basis for handling uncertainty in GIS.

It is a reasonable expectation that some combination of HW.model, AF.model and a formalization of HE.model is how SPP will eventually be solved. The integration of these approaches is most naturally accomplished within the context of expert systems since the decision of when to use each approach and how to use them requires a certain amount of expertise that must be formalized in a computer system.

HE.model is a particularly appropriate problem for the application of expert system technology. One of the first steps would be to categorize data layers according to their type of uncertainty, then take each layer separately and determine how to characterize the certainty of each arc of each polygon. These rules will be specific to each data layer. For example, parcel boundaries will have greater certainty, perhaps best expressed as evidential while a soil boundary may be most appropriately represented by fuzzy propositions.

There are a number of decision rules which need to be formalized in the development of an EPOP (expert polygon overlay processor). There needs to be a set of rules for specifying the relations between the layers. In terms of uncertainty, what special combination rules may be required. Also, given the existence of another layer use of that layer to provide additional evidence for making a decision. To handle SPP, an SPP identification procedure will need be formalized to detect SPP situations and prescribe a solution peculiar to that specific problem. There will be two dimensions to such solutions. One will be the geometry of the situation and the other is concerned with the specific data layers being overlaid.

4.2 Spatial Query

Spatial query and POP are not entirely independent operations for in order to answer many queries of a geographic nature, especially those of a multiattribute nature, there must be some POP either as a prelude to the query or as part of the query process. The question of whether all POP is conducted prior to a query or as part of a query process will not be addressed here as it has received considerable attention elsewhere. It is sufficient to state that we feel that the most reasonable solution will be a combination where infrequently needed relations are not preprocessed and stored while most frequently needed relations are subjected to POP and stored for ease of retrieval in later queries.

Our consideration of uncertainty in query will concentrate on spatial relations based on distance, direction, and extension. Spatial relations based on intersection and union of areas we relegate to the domain of POP. Much of the uncertainty regarding the definition of spatial relations such as 'near, 'far', 'northwest_of', can be traced to linguistic terms that are based on concepts with fuzzy boundaries. Thus, we will be looking at ways of incorporating AR into the spatial query process. Some fundamental progress has been reported in (16), however their efforts were limited to nearness and farness terms, and point-point relations. Each type of spatial relation will have to be formulated so that uncertainty associated with each can be represented and used effectively.

Of course, one of the main problems in using fuzzy spatial relations has been the requirement for acquiring membership values. To address this issue for some relations we anticipate incorporating SRAS in a similar form as reported by (16).

Once an answer has been found to a query there is the question of how to display the result to the user in such a manner that the uncertainty of the information is conveyed without the user having to also be an expert in ER and AR. We anticipate using an approach similar to that discussed by Farhoosh and Schrack (20). One drawback is that it may prove awkward to generalize to the monochrome domain. It is our feeling that the flexibility offered by color graphics is necessary for the meaningful display of uncertainty. This gets us into the domain of intelligent visualization.

5. CONCLUDING DISCUSSION

A number of significant problems need to be addressed before a comprehensive system is constructed for managing uncertainty in GIS. Like the general problem of expert system development, the acquisition of certainty/uncertainty measures is of fundamental importance. The experience of (16) suggests that uncertain multiperson concepts can be acquired, but that the representational framework will need to be in place before attempting knowledge acquisition. Therefore, we choose to develop our prototype in one of the more general representational forms - Horn clauses of Prolog which can be extended using support logic programming (12). The advantage here is in rapid prototyping rather than development of a very efficient commercial system.

There is also something to be said for the maintenance of multilayer geographic databases. Often the criteria of efficiency is based on a measure of a single overlay processing run, rather than the entire process of overlay update. We suggest that our approach will significantly decrease the entire overlay process as it is currently conducted. Likewise, the query process should be evaluated on how efficiently it completely answers an agency's query. Very often, a simple query must be run a number of times before a suitable answer is arrived at. There is also the case of queries that are directed at trying to construct an assessment of the certainty associated with the answer of a particular query. These are usually awkward, cumbersome exercises. Thus, by incorporating uncertainty within the entire query process the overall processing burden is reduced.

Another important aspect of expert systems is the ability to explain a line of reasoning. User's are likely to place more credibility in an overlay processor that can explain its reasoning than in the current "black box" procedural programming approach. Thus, development of an explanation facility for EPOP is an important component in managing uncertainty in a GIS.

In keeping with the spirit of the suggestion in (6), we anticipate gaining significant insight into POP and query processes through our attempt to build a small prototype which will incorporate uncertainty management. Such insight can contribute to the construction of GIS of greater utility. However, given the current level of support, it likely that several years will pass before the potential of this effort will be realized.

Currently work on the prototype EPOP and SPATQ are being conducted on the Department of Surveying Engineering's VAX 11/750 using Quintus Prolog as the primary programming language using interfaces to C and Fortran when procedural calls are necessary. The issues of developing suitable displays for depicting uncertain geographic information will be tackled using the same programming tools in conjunction with Tektronix 4208 color graphics terminal.

References

1. Robinson, V.B. and A.U. Frank. 1985. "About Different Kinds of Uncertainty in Geographic Information Systems,", Proceedings, AUTOCARTO 7, Washington, D.C.

2. Bedard, Y. 1987. "Uncertainties in Land Information Systems Databases," Proceedings Eighth Intern'l Symposium on Automated Cartography,pp. 175-184.

3. Robinson, V.B., A.U. Frank and M. Blaze. 1986. "An Introduction to Expert Systems for Land Information Systems," Journal of Surveying Engineering, 112(2):109-118.

4. Robinson, V.B., A.U. Frank and M. Blaze. 1986. "An Assessment of Expert Systems Applied to Problems in Geographic Information Systems," Journal of Surveying Engineering, 112(2):119-130.

5. Robinson, V.B., A.U. Frank. and H.A. Karimi. 1987. "Expert Systems for Geographic Information Systems in Natural Resource Management," AI Applications in Natural Resource Management, 1(1):47-57.

6. Robinson, V.B. and A.U. Frank. 1987."Expert Systems for Geographic Information Systems," Photogrammetric Engineering and Remote Sensing, in press.

7. Yager, R.R. 1986. "Toward General Theory of Reasoning with Uncertainty.I: Nonspecificity and Fuzziness," Intern'l Jrnl. Intelligent Systems, 1(1):45-68.

8. Shafer, G. 1986."The Combination of Evidence," Intern'l Jrnl. Intelligent Systems, 1(3):155-180.

9. Mangiaracina, S. and G. Beni. 1986. "On the Logical and Physical Combination of Evidence in Intelligent Machines," Intern'l Jrnl. Intelligent Systems, 1(2):143-152.

10. Maier, D. 1983. The Theory of Relational Databases. Computer Science Press, Rockville, MD.

11. Garvey, T.D., J.D. Lowrance, and M.A. Fischler. 1981. "An Inference Technique for Integrating Knowledge from Disparate Sources," Proceedings 7th Intern'l Joint Conf. Artificial Intell., pp. 319-325.

12. Baldwin, J.F. 1986. "Support Logic Programming," Intern'l Jrnl. Intelligent Systems, 1:73-104.

13. Tversky, A. 1974. "Assessing Uncertainty," Cognitive Psychology, 2:148-159.

14. Kosko, B. 1986."Fuzzy Knowledge Combination," Intern'l Jrnl. Intelligent Systems, 1(4):293-320.

15. Gaglio, S. R. Minciardi, and P. Puliafito. 1985. "Mulitperson Decision Aspects in the Construction of Expert Systems," IEEE Trans. on Systems, Man, and Cybernetics, vol SMC-15: 193-204.

16. Robinson, V.B. and R.N. Wong. 1987. "Acquiring Approximate Representations of Some Spatial Relations," Proceedings Eighth Intern'l Symposium on Automated Cartography, pp. 604-622.

17. Frank, A.U. 1987. "Overlay Processing in Spatial Information Systems," Proceedings Eighth Intern'l Symposium on Automated Cartography, pp. 16-31.

18. Guevara, J.A. 1983. A Framework for the Analysis of Geographic Information Systems Procedures: The Polygon Overlay Problem, Computational Complexity and Polyline Intersection, Ph.D. dissertation, Dept. of Geography, SUNY-Buffalo, Buffalo, NY.

19. Corson-Rikert, J. 1987."Updating Multi-layer Vector Databases," Proceedings, First International Geographic Information Systems Symposium, in press.

20. Faroosh, H. and G. Schrack. 1986. "CNS-HLS Mapping Using Fuzzy Sets," IEEE Color Graphics and Applications, June:28-35.

BLANK PAGE

SPATIAL TERMS AND SPATIAL CONCEPTS:
GEOGRAPHIC, COGNITIVE, AND LINGUISTIC PERSPECTIVES[1]

David M. Mark   Soteria Svorou   David Zubin
Graduate Group in Cognitive Science[2]
State University of New York at Buffalo
Buffalo NY 14260

ABSTRACT

Spatial terms, spatial concepts and spatial orientation represent a significant topic in geographic information systems research, and indeed in the development of spatial (geographic) theories. They are also an important topic in linguistics and cognitive science. Development of a "general theory of spatial relations" has been identified as a critical research topic on the research agenda for geographic information systems and geographic analysis. Thus far, however, it seems that research on this topic in the spatial sciences has been conducted in relative isolation from research in the cognitive and linguistic sciences. This paper first presents a general theory of spatial relations in language, based on cross-linguistic studies and informed by cognitive principles. Next, published research on the topic by geographers and other spatial scientists is reviewed. Finally, the paper identifies research topics in spatial language and spatial concepts which are critical to a general theory of spatial relations for geographic information handling and the development of geographic theory.

1. INTRODUCTION

Linguists and Geographers have a number of domains of common interest, the most recent of which grows out of their mutual participation with psychologists and computer scientists in Cognitive Science. This new common interest focuses on the human being's mental representation of geographic space and the linguistic structures into which this knowledge is coded in the communication process. Natural languages encode a variety of spatial information in their

---

1. This research was supported in part by NSF Grant No. IRI-8610517 to the third author and NSF Grant No.BNS-8318262 to Joan L. Bybee of the SUNY Buffalo Gramcats project.

2. Mark is a member of the Geography Department, Zubin and Svorou of the Linguistics Department. This paper represents collaborative research carried out in the context of the Cognitive Science Graduate Group.

grammatical and lexical structure (cf. Herskovits 1985; Vandeloise 1986; Talmy 1983; Zubin & Choi 1984; Zubin & Svorou 1984), and furthermore, the coding of space serves as a concrete basis for the more abstract representation of time (Clark 1973; Traugott 1978) and case role (Anderson 1972). Indeed, in the view of Lakoff and Johnson (1980), concrete spatial representation and its coding in language is a primary basis for all abstract concepts, and for all grammatical structure.

The interpretation of spatial reference in natural language is of particular importance for the design of natural language interfaces for geographic information systems (GIS). If GIS users could formulate queries in natural language, and rely on the computer system to correctly interpret the query, then GIS use could be greatly facilitated. A cross-linguistic perspective is important in GIS design. When systems are designed by English speakers, there is an imminent risk that the systems' query structures will not support linguistically represented spatial concepts of other languages. In fact, there is evidence that a speaker's underlying representation matrix for geographic information is bound to specific geographic locales, so that speakers of the same language from different locales might have conflicting underlying matrices. Some of this evidence will be reviewed here.

In this paper, we first will review briefly some spatial concepts of interest to GIS. Next, we will lay out the parameters along which spatial representation is coded in language, and then focus on a cognitive prerequisite for all spatial coding - the Reference Frame Problem - and its homocentric and geographic origins. Finally, after reviewing some research on the geometric "meaning" of spatial terms in a GIS context, we suggest some directions for future research which should benefit both linguistics and GIS.

2.SPATIAL REFERENCE IN NATURAL LANGUAGE

The range of spatial information coded in natural language includes POSITION (The Hilton is behind City Hall), DISTANCE (The Hilton is near the waterfront), DIRECTION OF MOTION (We walked toward the Hilton), PATH OF MOTION (We walked along the waterfront; down Delaware Ave.), the EXTENSION of objects (City Hall is a tall building) and spaces (Main Street gets wider as you go out of the city) and the SHAPE of objects and spaces (Main Street is curved; Buffalo is flat).

English speakers in Buffalo talk about position, for instance, when they use utterances like "The last subway stop is in front of the Aud". The positional information in this utterance is provided by the spatial expression in front of, but the actual interpretation of such utterances presupposes a cognitive model which goes far beyond the meaning of individual words. This cognitive model involves a figural object (subway stop), a ground object (Aud), a region-designator (front), and a reference frame integrating these elements.

The FIGURAL OBJECT (F) is the object which the speaker attempts to locate for the listener. The GROUND OBJECT (G) is a salient physical or geographic entity which serves as a point of reference. In the language user's cognitive representation of objects, each object has one or more REGIONS (R) (Miller and Johnson-Laird 1978) consisting of one or more of a) its interior, b) its surface, and c) an area adjacent to the object. For example, the representation for <ROOM> has an interior region, plus an exterior region adjacent to its access; <HOUSE> has an interior region, and is surrounded by an exterior region; <STREET> has a surface region and exterior regions along its sides. The size of an exterior region is problematic, since it is not bounded by the object itself. Instead, an exterior region has a SCALE determined by a) the size of the object itself, b) the distance to, and/or size of, other relevant objects, and/or c) a functionally-determined "use-space" around the object.

Furthermore, the exterior region of an object is partitioned in two ways. The first cognitive partition, based on the ubiquitously salient properties of gravity[3] and ground level, establishes a 1-dimensional vertical space (sub-regions above and below the object) and a 2-dimensional horizontal space around the object. The second cognitive partition further divides the horizontal space into four sub-regions corresponding to the bilaterally assymetrical front-back axis, and the symmetrical left-right axis of the human body (Clark,1973). Several types of evidence show that the human body is both the historical and the cognitive source for the 4-way partition. For example, in an extensive genetically stratified sample of the world's languages (Bybee 1984), most show evidence that terms for [FRONT] and [BACK] are derived from the corresponding human body parts (Svorou 1986). Many objects inherit [FRONT], [BACK], [LEFT] and [RIGHT] values for their horizontal sub-regions from the human body based on such principles as perceptual access, canonical encounter, and direction of motion. These canonical, or default, values are part of the object's cognitive representation. For a typewriter these are canonically fixed as the region intended to face the user ([FRONT]), its opposite ([BACK]), the region to the right hand of the user ([RIGHT]), to the left hand ([LEFT]), the upward region ([OVER]) and the downward region ([UNDER]).

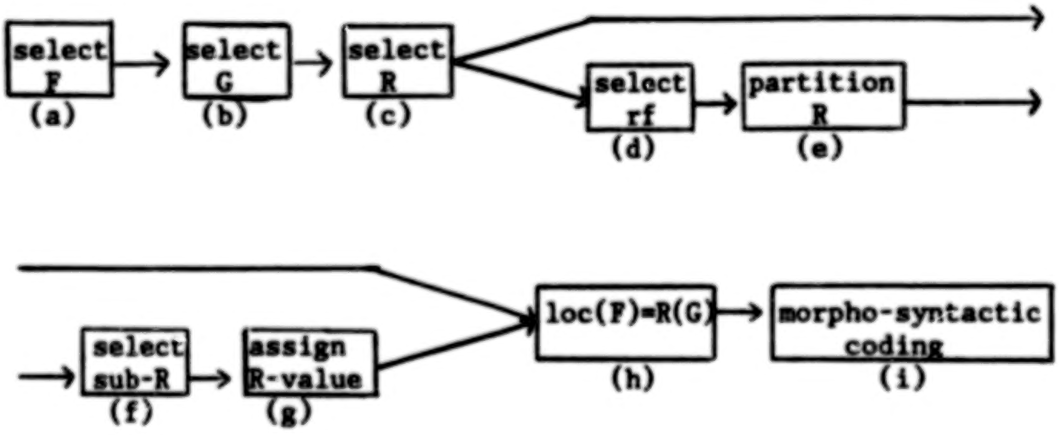
Finally, when a region is partitioned, as in the previous example, a

-----

[3] Not "gravity" itself as an abstract concept of physics, but rather the salient consequences of gravity: direction of falling objects, upright alignment of humans, trees, walls, etc.

REFERENCE FRAME (rf) is needed to assign values to the sub-regions. There are two types of reference frames: inherent and deictic. An inherent rf instructs the speaker/listener to use the default sub-region values for the specified ground object, which are "inherent" in the sense that they are coded in the cognitive representation of the object itself. We can find the inherent [FRONT] and [BACK] of a typewriter irrespective of the environmental setting, as noted above. A deictic rf instructs the speaker/listener to ignore the default sub-region values of the ground object (if any), and to seek values in its environment. These values are situationally (deictically) determined, rather than being inherent in the object. For example, the deictically-determined front sub-region of a table is the one facing towards an observer, while its back is facing away.

The following figure represents a possible procedural sequence for cognitive processes leading up to the linguistic coding of a spatial relation.



The speaker (or GIS) first (a) selects a Figural Object(F), perhaps in response to a query such as "Where is the last subway stop?". Next (b), s/he selects a salient Ground Object(G) with respect to which the position of F will be specified. Then (c), s/he selects a major-region(R) of G, which will be its inside, surface, or surround. If this region is sufficiently specific, the speaker can proceed directly to a linguistic coding of this information (h & i), using the semantic relation loc(F) = R(G), i.e. "the location of F is a region of G". This would yield a description such as "The last subway stop (F) is near (R) the Aud (G)".

However, if the speaker wishes to be more specific about the location of F, then (d) s/he must select a Reference Frame (rf), within which (e) the selected R is partitioned in four sub-regions (sub-R). Then (f) the speaker picks a sub-region, and (g) assigns it a R-value within the reference frame, for example, [FRONT] or [INWARD] or [SEAWARD]. The information is now ready for linguistic coding via (h) and (i).

If the speaker has selected an inherent rf, then the default value residing in the cognitive representation of <AUD> is selected in step (g), based e.g. on this building's main entrance, yielding a description such as "The last subway stop (F) is in front of (R) the Aud (G)". On the other hand, if a deictic rf is selected at (d), then a value is assigned at (g) to the same sub-region of G, but this time based on the observer's position, yielding a description such as "The last subway stop (F) is to the right of (R) the Aud (G)". Note that in all these examples, F, G, and R, but not rf, receive morpho-syntactic coding. In English and other languages we have investigated, reference frames tend to be inferred from the linguistic and situational context, and are rarely specified in the utterance. Complex region-designators such as "at the front of" (inherent) or "on the other side of" (deictic) not only specify an R, but also the rf used (Zubin & Choi 1984).

2.1 Sources of Reference Frames

Although reference frames are not typically coded in spatial expressions, there is evidence for their existence and diversity in the evolution of spatial terms. Natural languages evolve in a way which is consistent with the cognitive characteristics of their speakers. A cross-linguistic study of spatial terms based on a sample of 26 genetically unrelated languages carried out by Svorou (1986;1988) provides evidence for the sources of rfs that speakers of such diverse languages use in talking about position and direction. What is interesting about these findings is the overwhelming similarity of these sources across languages. Four different sources of rfs have been identified:

The human body. We have already pointed out that the 4-way partition of the surround of any object is inherited from the natural 4-way division of the human being's surround, which is based on the axes of the body. Quite naturally, then, it turns out that body-parts are used to designate such regions as [FRONT], [BACK], [ABOVE], [BELOW], [LATERAL] and [INTERIOR]. Body parts such as face, chest, forehead, mouth, eye, and head are used when talking about the [FRONT] sub-R. Back, buttocks, anus, loins are used to describe the [BACK] sub-R. Head denotes the [ABOVE] sub-R, whereas hips and bottom and foot describe the [BELOW] sub-R. Heart, flank, ribs and ear describe the two [LATERAL] sub-Rs, and stomach and blood the [INTERIOR] sub-R.

Each relation is specified by means of a set of body parts which are located on the same side of the human body. For instance, to talk about the [FRONT] region, body parts located on the front side of the human body (e.g. face) are employed. To talk about the [BACK] region, posterior body parts are used. Thus, people treat the position of objects the same way they conceive of their own bodies.

Canonical encounter. Human agents are perceivers and conceivers, participants in social situations and doers. We call the incorporation of human body and agent the "EGO". The physical and social limitations/possibilities on the interactions of EGO with other objects in the world define another set of reference frames that people use in the description of spatial relations. EGO interacts with other entities and other EGOs in specific ways. People normally face each other when talking, establishing a canonical face-to-face encounter of human beings with each other. The value [FRONT] can then be assigned to that side of any object which humans typically encounter when they use it. For example, we normally use objects with doors (houses, refrigerators, ovens, cupboards) interacting with the side with the door. Canonical encounter contributes, together with other sources (direction of motion), to the setting of the values of inherent regions to objects.

EGO-orientation. In addition to canonical encounter EGO interacts with ground objects in ways that are non-canonical. Non-canonical interaction can mean either that G has no inherent region values (like trees, columns,etc.), or that the current setting does not correspond to the canonical encounter. In these cases the situational orientation of EGO provides the reference frame for the assignments of values to the region of the ground object. Specifically, the language user must first establish a spatial relation between EGO and G. Then, the sub-regions around G are assigned values based on EGO's perspective. If G also has values assigned to its regions by canonical encounter, these values are overruled by EGO-orientation in non-canonical interactions.

Geographic. Human anatomy is the source of reference frames for the conceptualization of only certain spatial relations. Mostly, as we noted earlier, they are positional relations. People move, however, in the physical environment they live in and have to orient themselves within that environment. Their body plays an obvious role, but in a large scale environment other elements are required.

In the expression of direction the speaker's immediate environment plays a crucial role. In the derivation of reference frames, we have to date identified two types: grid-derived and goal-derived. A grid-derived rf is the familiar one with two orthogonal axes, found in geographical settings which present features to which these axes can be anchored. A less familiar grid system is radial, or polar, with directions defined toward or away from a center, and around the periphery. Grid systems seem to be based on salient geographic features, such as mountains, sea, coastline, river or other waterway. In contrast, a goal-derived rf requires that direction be specified in terms of an ultimate goal of travel, no matter what the local direction of movement may be. Goal-derived systems seem to be based on commerce rather than physical landmarks.

Grid-derived rfs. The Nicobarese, inhabitants of the Car Nicobar island of the South Andaman Sea, use a radial grid rf to talk about direction of motion, corresponding to the geography of their island. Two suffixes to verbs specify outward and inward motion. The suffix for "outwards" also means that the action of the verb is directed toward the seashore, and the suffix for "inwards" also means inland. These terms do not reflect a reorientation of the NSEW-based grid at all, but rather presuppose a radial spatial schema. "Inwards" is toward the unihabited jungle-covered center of the island, while "outward" is toward the inhabited periphery and the sea. In addition, a suffix meaning "up" designates to the left while facing the sea, and a suffix for "down", to the right, yielding directions along the circumference. The Nicobarese radial spatial schema is thus motivated by the radial organization of their geographic setting.

One does not have to turn to such exotic languages to find examples of the role of environmental landmarks in the language of spatial relations. The spatial descriptions of English speakers in Hawaii presuppose a radial grid rf in both casual (conversational) and formal (e.g. cartographic) contexts. They use <u>mauka</u> "mountainward" and <u>makai</u> "seaward" to designate direction along a radius and <u>`Ewa</u> and <u>Diamond Head</u> to specify direction along the circumference in Honolulu (Bier 1976).

At UC Santa Barbara, which is located on a strip of land between the sea and the mountains, <u>seaward</u> and <u>mountainward</u> are used as cardinal orienting directions in an orthogonal grid system. On the one hand, sea and mountains are omnipresent aspects of the environment; on the other, the coastline is diagonal to NSEW, making these global directions unsuitable.

Direction of flow in a waterway may provide the primary axis of an orthogonal grid system. Native Buffalonians talk about "going <u>down</u> to North Tonawanda" (which is north) and "<u>up</u> to Buffalo", sometimes puzzling non-natives. The Erie Canal, a focal feature in the economy of the region well into the 20th century, flows north out of Buffalo before turning east to Albany. Direction of flow of a waterway as an axis and direction specifier for grid systems is common among natural languages (Karok, Palantla Chinantec, Carapana, among others).

Another geographic element with similar function is a mountain or hill. Uphill and downhill directions in conjunction with the directions defined by the flow of a river provide a 4-way orientation system for communities that live in settings with such environmental characteristics (Karok). This also occurs in Freedonia, New York, where "up" is southward, which is up a hill and away from Lake Erie.

<u>Goal-derived rfs</u> Haugen's (1957) description of geographic directions in Iceland provides a prototypical example of goal-derived reference frames. The population of Iceland lives primarily in communities along the waterways of the coast. A typical fjord valley in the north east of the country uses <u>inn</u> "in" and <u>út</u> "out" for directions along

the fjord, and *auster* "east" and *suður* "south" for the orthogonal axis across the valley. Their orthogonal axes are thus in-out and east-south. The latter pairing is motivated by the actual location of neighboring communities in the next valleys. Although one travels northeast and southwest out of the valley, the final destinations lie almost due east and due south, respectively. This, in general, is the way direction is specified for local travel.

The island is divided into four conventional quarters called "North Quarter", "South quarter", etc. For long-distance travel from one quarter to another, the ultimate goal rather than the intermediate direction of travel is specified, so that E means going to the eastern quarter. This holds even in extreme cases: when traveling from the northwest coast (west quarter) to a location towards the north center, which belongs to the north quarter, you say you are traveling north even though your general direction is south (Einarsson 1944).

Goal-derived rfs seem to be used by departments of transportation in this country. For example, entrance signs to the NY Thruway (I-90) near Buffalo say "WEST" and "EAST", and state "Albany" and "Erie" as ultimate destinations (which do lie to the east and west, respectively), even though in the Buffalo region I-90 runs due N-S.

Global directions. The linguistic nature of global cardinal directions reveals more facts about the homocentric and geographical origin of reference frames. Cardinal directions are thought of as an absolute system of orientation based on the rotation of the earth and the magnetic poles of the globe. This knowledge, however, is not part of the linguistic origin of the terms for directions corresponding to the cardinal directions.

Communities assign absolute directions on the basis of environmental settings and celestial objects which are perceptually salient. The sun and its movement occupy a prominent position in the preferences of peoples. "The place where the sun rises" is the literal translation of the terms for East in many Indo-European languages (Buck 1949), and Navajo, Haka, Karok among others (Svorou 1988), while "the place where the sun sets" refers to the West in the same languages. Another celestial object defining a direction is the Big Dipper of the constellation Ursa Major, which occupies the northernmost part of the horizon. In Latin and Navajo the names of the constellation are used as designators of the North direction.

While East and West are predominantly anchored to the rising and setting position of the sun, the orthogonal directions receive their lexical content primarily from environmental elements such as the sea versus the interior (Guaymi), upriver versus downriver (Karok), or the location of major cities (Haka, Zuni) (Svorou 1988). Greek north and south come from the names of the northern wind for north and southern or rain wind for the south (Buck 1949).

Another frequent pattern of assignment of cardinal directions among Indo-European languages is with respect to a human observer facing the rising sun. "In front" is the east direction, "behind" is the West, "left" is the North and "right" is the South. In Avestan, however, the observer faces south when assigning orientation, so that "in front" is south, "behind" is north, "left" is east and "right" is west (Buck 1949). It is interesting to note that medieval world maps placed east rather than north at the top. This may be homocentrically anchored: the observer faces east, which is thus "in front", which then gets rotated to the top in a vertical projection.

3. RESEARCH BY GEOGRAPHERS AND OTHER SPATIAL SCIENTISTS

Peuquet (1986) has reviewed constraints on spatial queries within a GIS context. First, she divided the topic into spatial properties and spatial relationships. She claimed that all spatial relationships can be stated in terms of three primitives: distance, direction, and the boolean set operations (and, or, not). Peuquet went on to exemplify this claim, stating that, for example, "the higher-order spatial relation 'nearest neighbor' can be expressed as a series of relative distance relationships" (Peuquet 1986:461). Rather than focusing on the natural language aspects, however, Peuquet went on to develop a series of "spatial relationship operators" to be used in GIS database design.

In a series of articles taking a different approach, Vincent B. Robinson and his co-authors have studied the use of spatial terms in English. Terms such as "here", "near", and "in front of" are usually understood in natural discourse, and yet may represent very different geometric regions, depending on the context (reference frame and region scale).

Robinson and Frank (1985) identified two types of uncertainty in spatial data. One type results from the inability to measure or represent exactly characteristics or events which are inherently exact; the other deals with concepts which themselves are inherently uncertain or ambiguous. Much of the latter type of uncertainty relates closely to spatial terms in natural language. "Due to recent advances and its ability to represent and process the vagueness of natural language concepts, fuzzy logic is identified as [a] major area for future research in managing uncertainty in spatial information systems" Robinson and Frank (1985:440).

This idea was explored by Robinson and others (1985). They stated that Natural Language concepts "have been shown to be especially important (1) in the storage of geographic data as linguistic data; (2) as quantifiers and qualifiers in the data collection and data coding process; (3) as descriptors of the 'quality' of a geographic information product, such as a map; and (4) in the retrieval of geographic information from a geographic data base" (Robinson and others 1985:161).

Robinson and others (1985) used the PRUF (Possibilistic Relational Uniform Fuzzy) framework from fuzzy-set theory (Zadeh 1981) to provide representations of spatial and spatial-relational concepts embodied in words in the English language in a way that "preserves their approximate nature" (Robinson and others 1985:161). As an example, they attempt to define the concept embedded in the English word "near". Robinson and others (1985:162) presented an expression in PRUF that "represents a procedure that returns the fuzzy subset of [the set of all cities] comprising cities that are near [a specific city]". Basically, the method makes "educated guesses" of cities that may be at the boundary between NEAR and NOT_NEAR, until that boundary has been defined to a desired degree of certainty. They illustrate the process with a simulation which determines which points in a 9-9 grid which are "near" the point in the center of the grid.

Robinson and co-authors (Robinson and others 1986; Robinson and Wong 1987) continued this line of research. Five experimental subjects used an interactive program based on the PRUF method to identify towns which were "near" a target town. One test data set consisted of 29 settlements in Georgia, and the other of 112 places in Connecticut. The subjects could consult a 1:250,000 scale map during the sessions. One important extension of the method was an attempt to compare the NEAR sets for different subjects, or even for different target cities. Robinson and Wong also compared the sets resulting from several closely-related yet distinct terms: "near", "far", "in the vicinity of", "remote from", "close to", "distant from", "short distance from", and "long distance from". These terms led to similar yet different sets. The authors did not show how any of these sets could be predicted for a new set of places.

4. SUMMARY AND PROSPECTS

Spatial reference in natural language usually produces unambiguous communication among people, and yet the terms used often cannot be directly transformed into geometric relations for use in a data-base query. In linguistics, it has been shown that the reference frame is a key concept in interpreting such spatial referents. The fact that spatial references in 26 genetically unrelated languages all are based on similar principles suggests that those principles reflect the way in which spatial relations are held in the human mind.

Formal research by geographers and other spatial scientists has not been solidly grounded in theory or empirical results from the cognitive and linguistic sciences. The lack of a complete transformation from spatial language to geometry has been interpreted as a form of uncertainty, and treated using fuzzy set theory. The research has not been based on the reference frame principle, or other principles established in Cognitive Science.

Whereas it is clear that linguistic models of spatial reference in natural language are more advanced than research by geographers on this subject, both disciplines would appear to have the potential to profit from an interdisciplinary approach. For example, neither linguistic nor geographic research on this topic has explicitly recognized both the importance of the reference frame and the fact that the reference frame has a characteristic scale. For example, the statements "my barbecue is near my swimming pool,", "my house is near the University," and "Santa Barbara is near Los Angeles" all appear sensible, and may even be correct, even though the distances involved may be 10 meters, 5 kilometers, and 150 kilometers, respectively. Neither Robinson and others (1985, 1986) nor Robinson and Wong (1987) showed how they would accommodate the fact that the concept represented by the English word "near" depends on both scale and context. This scale effect is an important area for further research, both in the cognitive and linguistic sciences and as a part of the research agenda for GIS.

Finally, it is evident that a variety of specific reference frames exist, even among speakers of the same language. On the other hand, this variety seems redusable to a few general types. A computer-interfaced GIS should first be capable of determining the reference frame(s) used by a specific user before selecting the spatial terms to specify locations or directions.

References

1. ANDERSON, J., 1972: The Grammar of Case. New York: Cambridge University Press.
2. BIER, J.A.,1976: Map of O`ahu: The Gathering Place. Honolulu: The University of Hawaii Press.
3. BUCK, C.D.,1949: A Dictionary of Selected Synonyms in the Principal Indo-European Languages. Chicago: University of Chicago Press.
4. BYBEE, J.,1984: A Cross-Linguistic Study of Grammatical Categories. Proposal submitted and funded by the National Science Foundation.
5. CLARK, H. 1973: Space, time, semantics and the child. In T.E.Moore (ed) Cognitive development and the acquisition of language. New York: Academic Press. 28-63.
6. EINARSSON, S., 1944: Terms of direction in Modern Icelandic. Scandinavian Studies Presented to George T. Flom 37-48. Urbana, Ill., University of Illinois Press.
7. HAUGEN, E., 1957: The Semantics of Icelandic Orientation. Word 13:447-460.
8. HERSKOVITS, A., 1985: Semantics and pragmatics of locative expressions. Cognitive Science 9, 341-378.

9. LAKOFF, G. & JOHNSON, M. 1980: Metaphors we live by. Chicago: Chicago University Press.
10. MILLER, G.A. & JOHNSON-LAIRD, P.N., 1976: Language and Perception. Cambridge,MASS: Harvard University Press
11. PEUQUET, D.J., 1986: The use of spatial relationships to aid spatial database retrieval. Proceedings, Second International Symposium on Spatial Data Handling, Seattle, Washington, U.S.A., July 5-10, 1986, 459-471.
12. ROBINSON, V.B., and FRANK, A., 1985: About different kinds of uncertainty in collections of spatial data. Proceedings, Auto-Carto 7, 440-449.
13. ROBINSON, V.B., BLAZE, M., and THONGS, D., 1986: Representation and acquisition of a natural language relation for spatial information retrieval. Proceedings, Second International Symposium on Spatial Data Handling,Seattle,Washington,USA,July 5-10,1986,472-487.
14. ROBINSON, V.B., THONGS, D., and BLAZE, M., 1985: Machine acquisition and representation of natural language concepts for geographic information retrieval. Modeling and Simulation (Proceedings, Sixteenth Annual Pittsburgh Conference), Vol. 16, Part 1: Geography and Regional Science, 161-166.
15. ROBINSON, V.B., and WONG, R., 1987: Acquiring approximate representation of some spatial relations. Proceedings, Auto-Carto 8, 604-622.
16. SVOROU, S., 1986. On the evolutionary paths of locative expressions. In Proceedings of the Twelfth Annual Meeting of the Berkeley Linguistics Society. Berkeley: Berkeley Linguistics Society, University of California, 515-527.
17. SVOROU, S., 1988: A Cross-Linguistic Study of Spatial Relations. Dissertation in progress. SUNY Buffalo.
18. TALMY, L.,1983: How language structures space. In H.Pick & L. Acredolo (eds) Spatial Orientation: Theory, Research and Application. Plenum Press.
19. TRAUGOTT, E.C.,1978: On the expression of spatio-temporal relations in language. In Greenberg et.al. (eds) Universals of Human Language, Vol.III. Stanford: Stanford University Press.
20.VANDELOISE,C.,1986: L'Espace en Français.Paris:Éditions du Seuil.
21. YUHAN, A.H.,1988: Resolution of the spatial reference problem in narrative understanding. Dissertation in progress. Department of Computer Science, SUNY Buffalo.
22. ZADEH, L.A., 1981: Test-score Semantics for natural languages & meaning representation via PRUF. in Rieger, B., (ed.), Empirical Semantics. Brockmeyer: Bochum, 281-349.
23. ZUBIN, D.A.& CHOI,S.,1984: Orientation and Gestalt: Conceptual organizing principles in the lexicalization of space. In Testen,D., Mishra,V. & Drogo,J.(eds) Papers from the Parasession on Lexical Semantics. Chicago: Chicago Linguistics Society, 333-345.
24. ZUBIN,D.A. & SVOROU,S.,1984: Perceptual schemata in the spatial lexicon: A cross-linguistic study. In Testen,D., Mishra,V. & Drogo,J.(eds) Papers from the Parasession on Lexical Semantics. Chicago: Chicago Linguistics Society, 346-358.

DATA BASE MANAGEMENT

BLANK PAGE

TOWARDS THE AUTOMATIC GENERATION OF GIS DATA BASES FROM MAPS:
RESEARCH ISSUES AND STATUS

Barry J. Glick
Spatial Data Sciences, Inc.
McLean, VA 22101

ABSTRACT

Geographic information systems have a voracious appetite for intelligent geographic data including spatial and associated attribute information. Much of this information is currently encoded in hardcopy map form. The transformation of this information into a database representation suitable for use in GIS's is expensive, error-prone, and time-consuming. Advances in scanner technology, processors, pattern recognition and other hardware and software innovations hold promise for significant progress. This paper presents the major research issues that need to be addressed in order to achieve a cost-effective automated "map understanding" capability.

1. INTRODUCTION

Several factors have combined to create intensifying interest in the exploitation of raster-scanned map data. These include increased use of geographic information systems, the development of raster map data collections to support electronic map display systems, and cost/performance improvements in scanner hardware technology. In addition, progress in several closely related applications and technologies has stimulated the field. For example, the increasing popularity of engineering drawing document conversion systems (CAD conversion) has resulted in the development and improvement of software for vectorization, format conversion, raster and vector editing, and other functions related to engineering drawing conversion. Progress being made in image understanding / computer vision, character recognition /

natural language processing and computer processing architectures is also relevant to the problem of automatic data base generation from maps.

The goal of this technology is to create well-structured, tagged vector cartographic data bases from maps; hopefully within the same time frame it takes to scan the map (i.e. "on the fly" following Boyle, 1983). This objective has been much more difficult to reach than was thought by previous researchers and vendors. There exists a wide gap between these expectations and the reality in terms of system functionality, throughput, and required labor intensity. In reality, maps are imperfect from a scanning perspective, containing inconsistent symbology, breaks in symbology, noise, errors, and a variety of ways of encoding information that are difficult for an automated system to understand.

It is difficult to get a good picture of today's performance of map scanning and database creation systems because of the lack of experience with comparable systems in a production environment. The systems that have been used for production have had, to one degree or another, customized elements that makes comparison to other systems difficult. The use of scan digitizing of maps in a production environment has been most common in the terrain elevation application; i.e., scanning contour maps to produce, first, vectorized and tagged contours and, second (in most cases), digital elevation models (DEM's). The U.S. Forest Service (Liston & Gossard, 1985) reported throughput of 10 hours per map, where maps were scanned at a six mil resolution. This includes vectorization and editing but not DEM generation. Boyle (1983), on the Saskatchewan system, indicated throughput of 25 minutes per map (scanning at 250 lines per inch resolution) and Lebert and Olson (1982) using the SysScan product reported a throughput rate of one hour per map (scanning and vectorizing) and four hours per map for DEM generation.

2 THE PROBLEM: THE NATURE OF MAPS AS SCANNING SOURCE MATERIAL

There are a wide variety of maps that contain information of use in GIS applications. The characteristics of these maps have significant implications for the requirements for a map scanning / understanding system and therefore it is important to recognize these differences and their implications. Multi-use maps typically use some relatively standardized symbology and contain a wide variety of feature symbols and associated text and type. Special purpose maps, on the other hand, commonly employ some unique symbology and may include a restricted number of symbols and alphanumeric information.

Topographic maps are examples of multi-use and they utilize a large number of symbols, some which may be complex, representing a wide range of geographic features. Maps produced primarily to support road network route planning and navigation (road maps) will typically have a reduced set of features, with obvious emphasis on roads, associated travel facilities, and populated places. Because road maps are designed to be used by naive map reader, they employ map symbology which aids their use but may, through simplification, generalization, etc., make reconstruction of the "true" topology and geometry of the features difficult.

Thematic maps provide a large number of examples of special purpose maps. For example, single theme maps comprised of a polygonal tessalation of the plane (e.g., soils map) will typically be encoded using simple line symbology (i.e., smooth lines) and some form of area symbol which may be color fill, shading, or an alphanumeric code that corresponds to the attribute for the associated polygon.

Large-scale engineering and cadastral maps also place specialized functional requirements upon a scan-digitizing system. In these maps, which in some cases bear close resemblance to engineering drawings, dimensional and other measurement information appearing on the maps as type, is critically important to generating the desired output data base. In addition, specialized symbology is employed and the maps may have been generated in a relatively informal cartographic production environment resulting in loose control over graphic characteristics of the maps such as line thicknesses.

2.1 Map Symbology

Features represented on maps are encoded by means of symbol shape, size, pattern, color, and line style. Point, line and area symbols are used to encode a feature's spatial and attribute characteristics. The level of attribution ranges from purely nominal (in which are feature is simply identified or named), through various ordinal and categorical identifiers, to interval or ratio level attribution wherein a quantitative measure can be associated with the feature. In those cases when a feature is identified by an associated specific name or if there is interval / ratio level attribution, the information is usually provided by an alphanumeric label.

In point symbols, the spatial information provided is limited to feature position (center-point) only, although when area or linear features are portrayed by point symbols area or linear extent can be encoded by the arrangement of individual point symbols. For line symbols, the spatial information represented includes both feature position (center-line) and linear extent of the feature. In addition, line symbols are commonly used to represent the bounds of area features. Finally, area symbols portray feature position (e.g., perimeter) and extent in the two-dimensional plane. In a map scanning and vector database creation system, these spatial and attribute characteristics of the various symbols found on a map must be captured and encoded in the appropriate data representation scheme.

Type is a key component of map information. Using type, information concerning geographic features is encoded via style, size, color, and position. The contents of type include natural language words, numbers, special symbols and alphanumeric codes. The type may be positioned horizontally, along any straight-line orientation or angle, and sinuously to follow the shape of a linear or area feature. The type may be directly associated with a specific map symbol (point, line or area) or it may stand-alone and not have an associated feature symbol. The categories of type found on maps include feature names (generic or specific/proper), numeric values associated with a feature or characteristic (e.g., spot elevations) and grid/graticule notation.

Type handling requirements for a map understanding system include type identification (ability to distinguish type characters from other map information), font recognition, type recognition (recognition of individual characters/numbers), word/number recognition (recognition of word or number units), and word/number understanding (translation of word/number into semantic meaning). In addition, the recognized type must be associated with its corresponding feature symbol(s) if such a symbol is present. Meeting these requirements will permit the extraction of the attribute and/or feature information encoded via type and the use of that information to tag the corresponding feature.

2.2 The Role of Color

Color plays a key role in map information portrayal and can be viewed as both a major obstacle in a map understanding system as well as an important aid in feature recognition and map data processing. The importance of color for a map understanding is obviously a function of the source material available for scanning; when color separates are available, the color information is a priori available to the system. When the system must deal with color source maps, the system must be capable of identifying each color and tint used in the maps.

On typical color lithographed maps, solid colors, screened tints, and color combinations are found on the map. Even in simplified color schemes, the number of colors and tints can be quite large. For example, the USGS experimental scheme for topographic maps suggested by Gilman (1983) uses 16 distinct perceived colors / tints. Within a single color, several screened tints may be used. The relationship between features and type on the one hand, and color on the other, is not always straightforward. Type is typically found in several (usually solid) colors, depending on the associated feature class. Thus, text is not separable from symbology based on color alone. Also, multiple features may share the same color therefore limiting color's ability to determine feature separations. Indeed, one of the most important shortcomings of existing systems is the difficulty involved in distinguishing different feature found in the same color layer (Lee, 1985).

3. FUNCTIONAL REQUIREMENTS FOR A MAP UNDERSTANDING SYSTEM

The generation of GIS data bases from maps involves a series of required processing steps (see Figure 1); these steps may not need to be performed in a fixed sequence and the order that they are attacked will vary based on the particular technical approach used. The major stages of the process are depicted in Figure1. In this paper, a somewhat arbitrary grouping, ordering, and placing of each necessary functional requirement into a processing environment is used; please keep in mind this arbitrariness.

3.1 Scanner Data Processing Tools

The first set of required processing functions is comprised of tools to process the raw scanner output into a more useful, accurate, and efficient raster data base. These functions are good candidates for implementation within the scanner system itself and have proven amenable to hardware or firmware implementation. These functions include:

- automatic / adjustable dynamic thresholding
- 2-D compression and decompression
- noise filtering / removal

GIS DATABASE GENERATION FROM COLOR MAPS

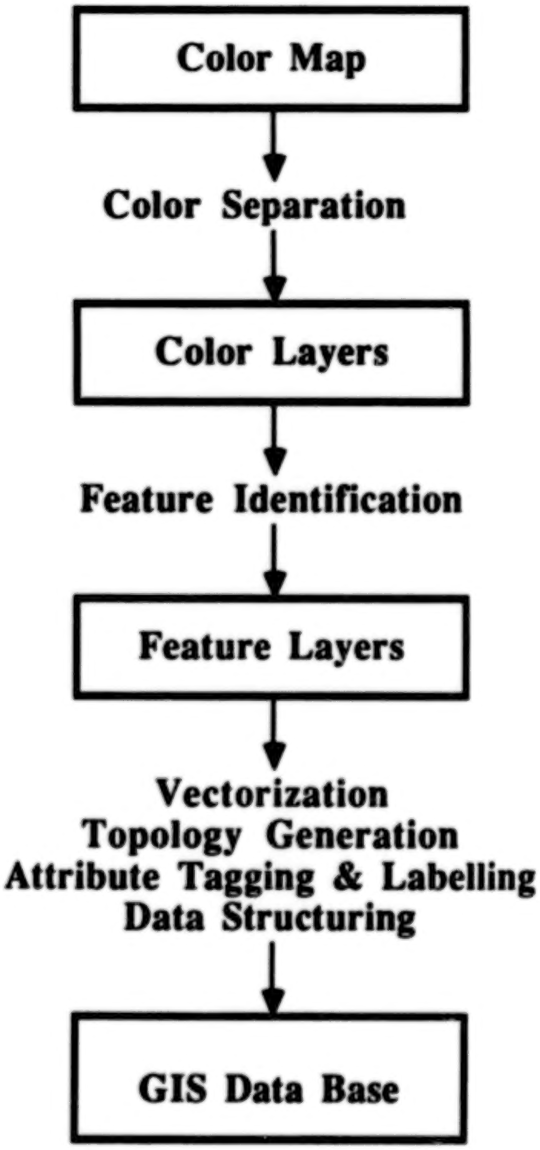

Color Map
Color Separation
Color Layers
Feature Identification
Feature Layers
Vectorization
Topology Generation
Attribute Tagging & Labelling
Data Structuring
GIS Data Base

Figure 1

- gray scale / color analysis, including histogram analysis (useful for thresholding as well), aids to generating color or gray-scale look-up tables needed to assign scanner response to color or gray shade "bins", and an interactive training set process to lead the system through examples of target colors and produce an appropriate look-up table.

The output of this stage of the processing consists of compressed, "clean" binary color separates and / or a multi-bit color composite which tags each pixel with the appropriate color code. Advanced color scanners (such as those produced by Tangent Engineering) can perform these functions and produce these outputs during the scanning process.

3.2 Raster Data Manipulation Tools

The raster data are then processed by raster data manipulation tools in order to:
1. Prepare raster data for a raster-based map storage and retrieval system, or
2. Prepare raster data for vectorization.

A comprehensive raster editing capability is required; the editing package must include the functions of state-of-the-art graphic arts and electronic publishing systems. When repetitive editing tasks are needed, an automated or aided approach should be available to the user. Other tools that may be necessary for this function include image processing tools (e.g., contrast stretch, edge enhancement, desharpening filters, etc.) and a raster formatting capability to allow for the introduction of the raster data to external systems. Raster data exchange standards, while lagging behind vector standards, are being introduced.

3.3 Processing of Separates

In the processing of the color separates, two major functions to be performed are continuity reconstruction and feature class identification and separation. In continuity reconstruction, gaps in symbols due to overprinting or masking are present in the color separates. Information on the color characteristics (or feature class if available) within "potential" gap areas can be used to determine the likelihood that the gap is "real" or is there because of the presence of an intersecting symbol or type. (Note - in some approaches, continuity reconstruction is addressed during the vectorizing process.)

In feature class identification and separation, type is identified within each color separate and is isolated or "demarcated" for later type processing. Feature class identification and separation involves the identification of particular feature classes within a single color layer. Depending on the granularity of the process, this may consist of, for example, separating transportation network features from buildings within the black color separate. The result of these processing functions is a set of feature and type separates which are homogeneous in terms of their symbology and type. This will greatly simplify subsequent processing.

3.4 Feature/Symbol - Specific Processing

During the next processing phase, each unique symbol / feature class is handled individually. However, to provide neighborhood context information, all feature information within the section of the map being processed should be available to each process. For line symbols classified as "simple" or "smooth", processing consists of skeletonization / line thinning, line extraction (vectorization) and topology generation (intra-feature). Complex line symbology is processed by use of symbol recognition and automated attribute tagging, center-line generation using, for example, a symbol-following algorithm, and topology generation. Alternatively, the complex line symbols may be skeletonized and vectorized first and then "de-symbolized" to create accurate center-line information. Comparison of these two approaches awaits development.

Point symbols are handled by means of a symbol recognition function and automated attribute tagging and position determination. Pattern detection tools are needed to identify linear or areal patterns of point symbols; in these cases the desired information is the center-line or perimeter of the feature and not the position of the individual point symbols.

Area features represented by true area symbology, e.g., color-fill or shading, require edge detection and delimiting polygon determination. Those area features that are symbolized by means of delimiting line symbology require line vectorization using either the simple or complex line symbology methods. For these area features, it may difficult or impossible to directly tag the features with area attributes without resort to type information. Finally, for area features represented by point or line symbol patterns, symbol recognition, edge detection and polygon / topology generation is needed.

Color-specific type requires font recognition, character/number recognition, word or number unit identification, and, optionally, word and number understanding. Positional information must be captured to identify the type's locational characteristics (including orientation, sinuousity, etc.).

3.5 Data Integration and Clean-Up

In this processing phase, the individual components of the map, the feature layers and the type layers, are brought together to create the integrated and tagged vector data base. As was true for raster data editing, a comprehensive and intelligent vector editing tool-box is needed, regardless of the degree of automation achieved. Feature attribute tagging and labelling is carried out by establishing type - symbol association using as much information as is available related to both the type and the feature to try to intelligently guide this complex task.

To create integrated topology across some or all of the feature layers, the individual layers are integrated and processed using one of several topology-generation algorithms. The level of topology captured, both in terms of extent across feature layers and level of redundancy, should be selectable based on the particular GIS application that the data will support. Geometric and topological consistency checks and anomaly detection routines are needed to assure the continuity of lines and the removal of unwanted spikes and gaps. Attribute consistency and logic checks are also needed to ensure proper attribution according to rules specifying the legitimate ranges of attribute tags associated with a specific feature.

3.6 Output Data Processing

Final processing functions required include those tools needed to create a seamless map data base with accurate geodetic or some alternate coordinate system. Map sheet panelling, boundary feature merge and dissolve and other tools are needed to create this seamless data base across a multiple map area of coverage.

Finally, the data base must be structured and formatted for easy introduction to a variety of GIS and map display and editing software packages. The system must support emerging standards for data exchange such as the Federal Geographic Exchange Format and the ISO 8211 / Spatial Data Exchange Standard.

4. CONCLUSION: RESEARCH AND DEVELOPMENT NEEDS

There is a clear need for research and development to improve the ability of raster scanning / map understanding systems to efficiently generate digital geographic data bases suitable for GIS use. One general category of research needs is the improvement of the robustness and functionality of processing tools using a knowledge-based approach. Many aspects of map understanding are amenable to a knowledge-based approach in which low-level raster operations and processing tools are intelligently combined with higher-level knowledge concerning the cartographic production process and the map reading process.

The identification and recognition of map symbols and type and the association between symbols and type are fertile areas for knowledge-based approaches. Other high-payoff applications for knowledge-based research include intersection / junction geometry generation (low-level approaches have difficulty in reconstructing the proper configuration of cased road intersections), continuity reconstruction following color separation, as well as the overall control of the map understanding process including selection of tools and processing paths.

The exploitation of new computing architectures and processing technologies is also a fertile area for map understanding research. As is the case for image understanding and computer vision, map understanding involves a large number of repetitive mathematical processing functions as well as higher-level reasoning and understanding functions. The use of parallel processors has been explored for some aspects of the map understanding problem (e.g., raster to vector conversion); with improvements in both parallel processing hardware and software, additional explorations seem warranted. The adaptive and cognitive aspects of the map understanding problem also suggest the possible value of exploring the use of artificial neural nets to handle these aspects of the problem.

BIBLIOGRAPHY

Boyle, A.R. (1983) The status of graphic data input to spatial data handling systems. in D. Peuquet and J. O'Callaghan (eds.) The Design and Implementation of Computer-Based Geographic Information Systems Amherst, NY. IGU Commission on Geographical Data Sensing and Processing.

Gilman, C.R. (1983) Map design at U.S.G.S.: A memoir. The American Cartographer Vol. 10, April, pp.31-49.

Langran, G. (1984) Map Design for Computer Processing: Literature Review and DMA Product Critique NORDA Report 72. NSTL, Mississippi. Naval Ocean Research and Development Activity.

Lebart, F.W. and Olson, D. (1982) Raster scanning for operational digitizing of graphical data. Photogrammetric Engineering and Remote Sensing Vol. 48, pp.615-627.

Lee, M.P. (1985) The Scitex raster graphic processing system.
in Proceedings of the Workshop on Methods and Techniques for Digitizing Cartogtaphic Data Open File Report 85-307. Reston, Virginia. U.S. Geological Survey.

Liston, R.L. and Gossard T.W. (1985) Line Trace overview.
in Proceedings of the Workshop on Methods and Techniques for Digitizing Cartogtaphic Data Open File Report 85-307. Reston, Virginia. U.S. Geological Survey.

Peuquet, D.J. and Boyle, A.R. (1984) Raster Scanning, Processing and Plotting of Cartographic Documents Williamsville, NY, SPAD Systems, Inc.

BLANK PAGE

Map Algebra: an object oriented implementation

Kelly K L CHAN
Denis WHITE

Laboratory for
Computer Graphics & Spatial Analysis
Graduate School of Design
Harvard University

48 Quincy Street
Cambridge, MA 02138
U.S.A.

Abstract

Object oriented programming is an effective software development paradigm. However, to seriously harness this implementation tool, we need a precise conceptualization of the fundamental application. The specific spatial modelling application we considered in this paper is the map overlay methodology, one of the traditional land use planning techniques.

From the his own Map Analysis Package, Dana Tomlin, in 1983, derived "Map Algebra." The algebra defines a set of spatial operations which formalizes the "map overlay" planning methodology. An object oriented implementation of Map Algebra is reported here.

0. Introduction

Object oriented programming has received a great deal of attention recently; discussions can be found in ***Smalltalk*** literature, proceedings of ***OOPSLA*** and ***Objected Oriented Databases Systems***, in ***SIGPLAN*** October, 1986, etc. A precise definition of object oriented programming is difficult to come by. Data abstraction is one of the most significant concepts in object oriented programming: complex problems can be abstracted into interacting components and a programmer can defined data types for these component objects. This object based programming paradigm was first realized in ***SIMULA*** during the 1960s; according to Nygaard, the concept of objects was actually started in the 1940s for the construction of simulation model of complex systems. In "Basic concepts of object oriented programming" Kristen Nygaard provides a statement of object oriented programming:

> In object oriented programming an information process is regarded as a system developing through transformations of its state. The substance of the process is organized as the system components, called objects. A measurable property of the substance is a property of an object. Transformations of state are regarded as actions by objects (Nygaard, 1986, 131).

Many object oriented programming languages have appeared since ***SIMULA***; most notably ***Smalltalk***. The language being used in this paper is ***C++***, a general purpose programming language that grew out of the ***C*** programming language. A class in ***C++*** is a user-defined type that matches more closely concepts of an application then the fundamental bit-byte-word data types; thus, it reflects more effectively the programmer's conceptualization of the problem.

Whereas ***Smalltalk*** presents a complete object oriented programming environment, ***C++*** was designed to be used in a traditional compilation and runtime environment such as that provided by ***UNIX***. Hence, ***C++*** is usually categorized as a hybrid object oriented programming language. Some of the advantages of object oriented programming, say, the ease with which a programmer updates an application in ***Smalltalk***, are in fact time consuming and rather tedious tasks in ***C++***. Nevertheless, the hybrid language provides a comfortable platform to migrate into object oriented programming from the ***C*** programming language.

Besides the benefit of data abstraction, there are also a number of other important aspects in object oriented programming. The most often cited advantages of object oriented programming language are: data hiding, inheritance and dynamic binding. Notions of object oriented programming language and programming paradigm are fully discussed in (Birtwistle 1973, Goldberg 1983, Nygaard 1986 and Stroustrup 1986).

The aim of this paper is not as much to introduce the concept of object oriented programming as to demonstrate the need for basic research in geographic information processing if we are earnest to benefit from advances in the computer sciences, such as object oriented programming and database management.

Object oriented programming is an effective programming paradigm. However, to seriously harness this implementation tool, we need a precise conceptualization of the fundamental application. Spatial modelling, the representation of phenomena over space, is the application we are interested in. The specific spatial modelling application we consider with the object oriented paradigm here is the map overlay methodology, one of the traditional land use planning techniques. The next section provides an excerpt from Steinitz's review article (Steinitz, 1976) on hand-drawn overlay and describes the overlay methodology in general.

Steinitz's article on overlay was partially a summary account of early experiments on computer-assisted overlay models. These early computer-assisted overlay models were essentially descriptive in nature; the procedures were but a direct implementation of manual operations found in hand-drawn overlay. Recent pursuits of prescriptive analysis using the overlay methodology demand a more rigorous evaluation of the methodology itself and, perhaps, a strategy to incorporate extensions.

One of those earlier computer-assisted overlay experiments culminated in the Map Analysis Package (MAP). The second section in this paper presents the Map Analysis Package of Dana Tomlin, some background information about his "Map Algebra" and also its formal definition. Concepts of the map layer, together with other MAP operators, are introduced as symbols in the algebra. The algebra and an implementation example are included.

The third section reports an implementation of MAP using *C++*. We also demonstrate the increased flexibility of the object oriented implementation over the original MAP and its language interface. We argue that while the original language interface is extremely useful to a novice user, the object oriented classes and methods are, however, well suited to experienced users who intend to derive more complicated systems from the algebra.

1. Map Overlay Methodology: History

The map overlay methodology is essentially a technique, in environmental planning, to summarize many spatially distributed variables. In ***Town and Country Planning Textbook*** where Tyrwhitt delineated the scope of a planning survey in the chapter "Surveys for Planning," she also described, almost as an adjunct, how such surveys should be recorded; her discussion, Steinitz suggested, was the first explicit explanation of the map overlay methodology.

> As far as possible maps should be drawn on transparent paper, so that when completed the maps to the same scale can be "sieved" - i.e. placed one on top of another in turn so that correlations or their absence can be noted. Where relevant "Sieve maps" should be made bringing out the degrees of correlations noted on certain matters. Only when this work is complete can the plan be formed, for, until then, many important factors in determining the lines of the plan will not be brought into focus (Tyrwhitt, 1950, 157).

In the 1960s, Ian McHarg further popularized the overlay methodology by a number of planning studies that were based on an overlay process; these included the Staten Island Study, published in ***Design with Nature***. Later on, a 1962 study by Christopher Alexander and Marvin Mannheim at the Civil Engineering Systems Laboratory, Massachusetts Institute of Technology ***The Use of Diagrams in Highway Route Location: An Experiment***, demonstrated the analytic strength of the methodology in dealing with a rather complex locational problem. Steinitz (1976, 447-448) described the three particularly important elements of this study in relation to development in the map overlay methodology:

> Twenty-six different diagrams, each depicting the relatively "utility" or desirability of the highway location for one specific requirement of the highway, are considered for analysis. The study recognizes that not all

> factors can be geographically overlaid at one time, and proposed a formal procedure "tree" for the combination and weighting over of overlays....
>
> This study surpasses traditional overlay applications in several ways: in its use of an explicit weighing system for the combination of different components, in the idea of superimposing the different diagrams or maps photographically as separately timed exposures on one print, and in the procedure used to structure the synthesis of the 26 different diagrams.

In his summary remarks, Steinitz reiterated the traditional and fundamental aspects of the map overlay methodology in environmental planning. Although he found, no doubt, the applications at the time of his study had become more complex, diverse and technically sophisticated, Steinitz stated that the use of the overlay approaches differed little between their early development and the applications then.

> ... it is clear that the basic methodology and the underlying logic have changed little. We combine data maps of soil, slope, and other elements in the same manner that Warren Manning probably did in 1912. We commonly hand draw each data element as a complete separate maps through overlays - visually, by redrawing or by photographic reproduction - into some sort of composite representing the analysis (Steinitz, 1976, 449).

However, the concept and the approach in map overlay have changed in the 1980s indeed. Ian McHarg had applied the methodology in a manner quite unlike the earlier applications; he had computed various summary statistics, including some cost-benefit measures, for his plans in entirety and for individual components of those plans. These exercises by McHarg offered the obvious opportunity of extending the map overlay methodology beyond its original, purely visually oriented descriptive functions into the truly analytic, prescriptive realms. The drawing and redrawing of zones on the overlay, not just the overlaying of transparent maps; the generation of design alternatives, not just the evaluation of alternatives can be done with a combination of design intuition and analytic thoughts.

1.1. Computerized Map Overlay

To realize the map overlay methodology on a computer is relatively straightforward and has been a rather popular activity. Computerized map overlay is sometimes called "data overlay" to distinguish it from the physical optical processes. Tomlin also refers to it with the more generic term "cartographic modelling" which he defined as a form of information processing involving the synthesis of mapped data.

Two of the more sophisticated overlay packages are available at GSD; they are: Map Analysis Package (**MAP**) by Dana Tomlin (Tomlin, 1983a) and **Odyssey** from the Laboratory for Computer Graphics and Spatial Analysis (Morehouse and Broekhuysen, 1982). Although both perform overlay analysis, the two differ significantly in terms of data structure and program complexity. Actually, to make **Odyssey** functionally equivalent to **MAP** in map overlay operations, we need to add to it a fugitive geographic processing module called **Harmony**. **Harmony** is developed by Denis White (White, 1983).

Beyond GSD, the applications of computerized overlay process in planning is best summarized in a book by Eric Teicholz and Brian Berry (1983) ***Computer Graphics and Environmental Planning***. Three other recently published reports are: "The Map Analysis Package as the core of an integrated water resources information system," South

Carolina Water Resources Commission by Christopher Heivly (1985); "Geographic information system technology as a tool for ground water management," University of Vermont, School of Water Resources by William Hendrix (1985); and"Soil erosion planning in Wisconsin," University of Wisconsin/Madison, Land Information and Computer Graphics Facility by Nick Chrisman (1985).

1.2. Prescriptive Analyses in Planning

Prescriptive analysis in planning can be divided into three parts: the generation of the alternatives, the description of the alternatives and the evaluation of the alternatives. Geometric optimization problems are so complicated at times that planners routinely have to reduce the solution space, rather intuitively, simply to make the problems bit more manageable in the descriptive sense. Research in prescriptive planning is to explore means to integrate or, at least, to support intuitive designs with analytic thoughts in environmental planning. What McHarg had done with the map overlay methodology in the 1960s clearly demonstrated that prescriptive planning with map overlay is attainable.

We have, thus, a mature planning technique, the map overlay methodology. Also, we have demonstrated the adaptation of this mature methodology to a new technology, computerized map overlay or cartographic modelling. Yet, to research the new opportunities delineated by McHarg's works, we can no longer depend on simply translating the proven manual procedures into modern forms; we do not have an old method to mimic any more.

2. Map Algebra

A small taste of the Map Analysis Package (MAP) as a cartographic modelling system before we venture into the discussion of map algebra. Bits of history. MAP was originally implemented by Dana Tomlin as a subsystem of IMGRID (Sinton, 1979, and Tomlin, 1986). IMGRID was designed by David Sinton at GSD in the late 1960s; it is the successor to GRID, developed by Steinitz and Sinton in 1968, which most likely was the grandfather of computerized map overlay systems. The Tomlin subsystem of IMGRID is the analytic modules that later evolved into MAP. There are now numerous versions of MAP, including a commercial incarnation called PMAP, and MAP type functions are found in a number of systems.

This section begins with two language examples from the Map Analysis Package. Followed by a summary Tomlin's opreator set. A formal definition of Map Algebra is also presented.

2.1. Map Analysis Package language examples

The following two examples are taken directly from the MAP programmer manual; they have been, of course, simplified. These two examples are nevertheless representative of the MAP programming language and they also illustrate a significant concept in MAP: classes of spatial operators. It will be sufficient, for the simple examples at this time, to ignore the implementation trivia and to consider an overlay, a "sieved map" drawn on transparent paper, as Jaqueline Tyrwhitt explained, only as a matrix in linear algebra. In fact, this is precisely how an overlay is represented by MAP inside the computer program, obviously, with bits and pieces of other data processing stuffs that no one really cares about much. The ADD instruction adds two such matrices, element by element, and puts the results on a third. This is the first class of instructions in MAP; they treat elements in a matrix as if they are totally independent of one another.

ADD overlay$_1$ TO overlay$_2$ FOR overlay$_3$

- The ADD operation creates an output overlay by adding the values of two input overlays on a point-by-point basis.

RADIATE is of the second class of operations in MAP and it looks at the geometric dependency among elements on a matrix. Let assume, for the time being, we have a metric defined as a function of the indices on the matrix. The first class of operations is called point operation. This second class is called neighbourhood operations. There is also a third class but we save that for later discussions; it is called region operation. The geometric dependency in RADIATE deals with the straightforward, everyday concept of visibility of an object from various locations. It is rather complicated computationally. The syntax of the instruction and the explanation are transcribed below.

RADIATE overlay$_1$ AT height TO distance OVER overlay$_2$ ONTO overlay$_3$ THROUGH overlay$_4$ SCREENING diameter ABOVE angle FOR overlay$_5$

- The RADIATE operation creates an output overlay by delineating the viewshed of selected points on a three-dimensional surface. A viewshed is the set of all points on which objects of specified heights are visible from one or more of the viewpoints selected. Visibility is determined according to whether or not an unobstructed line of sight can be traced from a specified height above the center of a viewpoint to the height of an object above the center of that point. Lines of sight may be obstructed by the three-dimensional surface itself and/or barriers of specified width and height above it. The designation of viewpoints, the heights of both objects and barriers, and the position of each point on the three-dimensional surface are defined by input overlays. Viewpoints are defined on overlay$_1$ for points with a value greater than 1. Overlay$_2$ is the three-dimensional surface. The objects of interest are defined on overlay$_3$, and the barriers on overlay$_4$.

The two examples show the general structure of the MAP language. Every instruction in MAP begins with a verb that specifies the operation; e.g. ADD and RADIATE. The verb in the instruction is followed by a number of objects and qualifiers. The objects in the examples are overlay$_1$, overlay$_2$, ... overlay$_5$, height, distance, diameter and angle. The qualifiers can be the general ones such as TO and FOR, but sometimes are operation specific, as in RADIATE: AT, OVER, ONTO, THROUGH, SCREENING and ABOVE. The verbs or operations in MAP can be summarized into five categories: the data management group, the display group, the program control group, the system INSERT instruction allows users to augment the original system with customized procedures, and finally, the data analysis and manipulation group from which the two above examples are drawn.

2.2. Map Algebra Concepts

In 1983, Tomlin presented a paper at the Harvard Computer Graphics Conference, "A Map Algebra," with the aim to establish a starting point for coordination on the development of digital overlay mapping techniques. Using the language system from his Map Analysis Package, he defined the set of operators and operands for cartographic modelling, an algebra of sort.

The Map Algebra described in Tomlin's paper is focused on the data analysis and manipulation operations inside MAP. Although data storage and management, not to mention data display routines, have traditionally been the centre of attention in geographic information processing research, we are, for the most part in this discussion, concentrated on data manipulation operations only.

2.2.1. map overlay metaphor

Why write an algebra of map overlay?

The map overlay planning methodology has a long tradition in environmental planning, the significance of a map overlay metaphor is apparent. Sinton explained his design objective as follow (Sinton, 1978,2):

> Retaining schemes to teach people how to use information systems and geographic data will only begin to generate such users if the necessary skills can be conveyed speedily and simply. Therefore we must make such systems responsive to the needs and the experience of such users. More precisely this means making the data structures simple to comprehend and replicating the analytic procedures traditionally used by such professionals. ...

Systems such as IMGRID and MAP are designed as map overlay metaphors. They are realized specifically to manipulate spatial information according to processes that mimic the conventional manual overlay methodology in environmental planning, although these processes are specified in some internal formal language of the systems. Given our experiences with formal map overlay languages, the purpose of an algebra of map overlay is to further the examination of intuitive thoughts in those domains.

2.2.2. cartographic modelling versus geographic modelling

The map overlay approach is a cartographic modelling methodology. In cartographic modelling the model, by and large, represents a study area with single or multiple graphic images of the area. All the objects on the image are addressable by their position on the image; i.e., we implicitly label objects by their physical position on the image. Every object, be it regular or irregular geometrically, carries with it certain characteristic value: elevations, reflected radiation, soil type, etc.

On the other hand, the model in geographic modelling represents a study area by the spatial relations among the many geographic entities found within the area. Entities in geographic modelling are named according to their types; we label these entities explicitly by either a proper or, more often, a generic name: point, polygon, elevation monument, watershed, Gund Hall, etc. Their positions within the study area are depicted in but one of the many spatial relations they are allowed. It is important to note, then, cartographic modelling should not be considered as anything more distinct but that of a special case of geographic modelling. There is, though, one significant constraint: spatial objects in cartographic modelling are geographically independent and all belong to the same type.

2.2.3. grid cell data structure

Objects in cartographic modelling are identified, implicitly, by their positions within the study area. Hence, it is sometimes efficient to control the locational designation of these objects through the imposition of an imaginary grid over the area, an implementation artifact. MAP, for example, has a grid cell data structure. As Chrisman had pointed out in "Fundamental principles of geographic information systems" (1987), earlier data structure designs in geographic information systems were essentially driven by technology instead of well founded theories. Although the semantics of certain MAP operators do assume an underlying grid cell gemoetry, the subsequent algebra as developed by Tomlin or in here is not restricted to any such geometry.

The values associated with these grid cells can be interpreted in two different ways: the value can be a complete description of the area of a grid cell, or the value can represent a point sample of a particular characteristics in the grid cell. Conventions must be developed to interpret the aspatial attributes for a cell and also its spatial attributes. Similar in any modelling exercise.

2.2.4. operator sets

In MAP all operations are on its data elements - map overlays. The cell data are defined on map overlays. A data element in MAP is precisely that of a sieved map drawn on transparent paper as Tyrwhitt described 35 years ago, or of a separation in a conventional paper map. The operations in the computer systems allow us to, literally, outline and colour code zones on the map, to redraw and to recode, and to overlay the many map separations. A complete map is composed of many separations; in cartographic modelling, a complete model is composed of many such data elements except we can manipulate each element and combinations of elements at will and with ease.

An operand in Map Algebra is a map overlay. Operators are categorized into three classes; the three classes are: point, neighbourhood and region. Neighbourhood and region operators are functions on spatial relations; the point class functions are essentially aspatial.

Peter Burrough in his recently published book ***Principles of Geographical Information System for Land Resources Assessment***, reworded some of the function names, expanded the set of operators with a few image processing type operators and, perhaps more importantly, introduced interpolation operators. Few of the changes are significant, however, they are included in our discussion when warranted.

2.2.4.1. Point characterizing

The point characterizing functions are the most straightforward of the three. Essentially, these functions take the ordered sequence of points on a set of maps and operate on each point in turn. Point characterizing functions, by and large, treat points, the operands, as geographically independent. In other words, these are aspatial functions. Among the MAP operators that compute new values on a point-by-point basis are: **recode** and **cross** assign new values according to numerical criteria or as cross tabulation with other layer, respectively; **add**, **subtract**, **multiply**, **divide**, **exponentiate**, **maximize**, **minimize** and **average** return for each point a computed value from two or more overlays; **copy** is self explanatory; **cover** is copy with transparency; and, finally, **slice** stratifies a variable on an overlay.

2.2.4.2. Neighbourhood characterizing

If we consider point characterizing functions as functions that operate on matched ordered sequences of points on a set of maps, neighbourhood characterizing functions operate on a set of points defined by some neighbourhood forming function, mostly, on a single map. Commonly used neighbourhood forming functions are (i) connectivity, i.e., all adjacent cells to a limit point, and (ii) distance criterion, i.e., all cells within a certain distance from the limit point. Operators in this class include: **scan** computes summary statistics for each neighbourhood; **differentiate**, **orient**, and **profile** measure surficial slope, aspect and cross-sectional characteristics, respectively; **spread** measures proximity; **radiate** delineates viewsheds; and **drain** simulates surficial flow. Burrough added two more functions: **trend** and **optimal**. Both operators perform interpolation while **optimal** does the kriging interpolation over a neighbourhood using the global variogram, **trend** fits a least-square surface to the entire map.

2.2.4.3. Region characterizing

The region characterizing functions somewhat bridge the gap between cartographic modelling and the requirements in spatial analysis. MAP and Map Algebra do not have a "region" primitive as such; the concept of "region" is realized indirectly by the label given to each point on the map. This inheritance, propagation rule is not an inelegant method to surmount some of the common dimensionality problems in geographic information processing. Although these region characterizing functions do not actually realize spatial relationships, they allow an effective surrogate in the Map Analysis Package for quasi-geographic modelling.

Region characterizing operators include: **score** the regional equivalent of **scan**; **size** measures the area of regions; **euler** and **survey** characterize regions according to their topological characteristics and their shape; and **span** measures the width of a region across a specified geographic point.

The following table summarizes the three classes of operators in MAP. Arithmetic operators are the familiar natural number arithmetic. Surficial operators under the class of neighbourhood operations include the calculation of slope, normal and azimuth.

point	neighbourhood	region
recode/cross	scan	score
arithmetic ops.	surficial ops.	size
cover	spread	euler/survey
copy	drain	slice/sort
	radiate	span
	interpolate ops.	

3. Map Algebra Specifications

In his 1983 paper "A Map Algebra," Tomlin explored the notion of constructing a formal algebra for cartographic modelling and proposed a set of map overlay operators as an initial point of the research. He demonstrated with ease that a simple algebra can be written using nothing but the data manipulation language from his Map Analysis Package. The following is a partial definition of Map Algebra. Full specification is in Appendix A; an implementation example of the specification can be found in Appendix B.

In the definition of Map Algebra, we have chosen to define the types according to the established concept of data element in the map overlay methodology, i.e. map overlay. We define three map overlay types: (i) the basic map overlay, (ii) the topographic map overlay which supports relief operators, and (iii) the thematic overlay which handles operations on 'data encoded' regions. We also have two additional data types: geographic point is essentially one named element on the matrix for any of the three map overlay types; and pSet is a group of connected geographic points, it is called "clump" in the Map Analysis Package language.

3.1. Symbols and Primitives

The method of specification follows those of Guttag (1978). The specification method of Guttag is called aglebraic specification and it is consisted of two parts: the syntactic specification and a set of axioms. For the purpose of an example, we define a type called "map layer"; the type "map layer" is the type of interest. Type "map layer" is defined on "geogrpahic point". In the example we also use three other types: topographic layer, geographic point and real.

A "map layer" is the most elementary data element in MAP; it contains the map image, the "dimensions" of the layer and other documentary items. The map image is composed of locational points; points defined on the image are of type "gPoint." Groups of points are "pSets."

A "topographic layer" represents the study area as if it is a continuous surface. "Neighbour- hood" is defined on topographic layer as piecewise continuous surface element of the surface which covers the entire study area.

Type map_layer

set: map_layer, topographic_layer, geographic_point, real

3.2. Syntax

The second part of the specification defines the syntax for the functions of the type. Here we have four such functions. POINT sets the value for a geographic point on a map layer. ADD adds two layers. COVER is what generally referred to as "copy with transparency;" i.e. values of the first layer is copied onto the second wherever such values satisfy a selection criterion, usually "not equal to zero." SPREAD draws the minimal spanning tree from a geographic point over a terrain.

The function POINT has an added significance. In this example, it is in the "reduced set" of our data type map layer. To ensure that a specification is complete, or more accurately, sufficiently-complete, the behaviour of all functions must somehow be derivable from this reduced set. The construction of a reduced set is a heuristic for axiomatization based upon the understanding of the application domain, the completeness of an alegbra is, of course, undecidable.

syntax:

function POINT	map_layer × geog_point × value	→ map_layer
function ADD	map_layer × map_layer	→ map_layer
function COVER	map_layer × map_layer	→ map_layer
function SPREAD	topo_layer × geographic_point	→ map_layer

3.3. Axioms

Finally, the specification must define the legal behaviour of the functions. Before we consider our specific specification, there are two implicit axioms that must be present in any specification. The first such implicit axioms states that a function returns "error" if any of its arguments does not belong to the domain of the function. The second axiom states that a function returns "error" if any of its arguments is itself "error."

The ADD and COVER functions are quite straightforward. For SPREAD, we have referred to the function INTERPOLATE and neighbourhood. INTERPOLATE is a hidden function; it returns a topographic layer which guarantees to be continuous within the neighbourhood surrounding a geographic point. The function is not part of the algebra and serves its purpose in the specification only.

We can read the SPREAD specification as follow. For any topographic layer, any geographic point or any range; SPREAD first collects all the points within a neighbourhood of the geographic point that satisfies the range requirement. For points in the newly created set that cost us less to connect to then previously possible or for points that have never been considered in the operation, SPREAD puts a new cost value for each of the points on the output map layer, and SPREADs from them in turn.

```
axioms:


A,B,C ∈ map_layer, R ∈ topographic_layer,
g, γ ∈ geogreaphic_point, Nb ∈ neighbourhood,
and $v_1$, $v_2$, value, range ∈ real

ADD( POINT(A,g,$v_1$), POINT(B,g,$v_2$)) = POINT(C,g, $v_1 + v_2$)

COVER( POINT(A,g,$v_1$), POINT(B,g,$v_2$))
        = if ($v_1 \neq 0$) then POINT(C,g,$v_1$) else POINT(C,g,$v_2$)

 SPREAD( INTERPOLATE(A,Nb,g), g, range))
        = ∀(γ) (γ ∈ Nb)
        [    cost = g.value+distance(γ,g)
             if ( (cost ≤ range) and ((γ.value > cost) or (γ.value = null)) )
             then    POINT(m,γ, g.value + distance(γ,g));
                     SPREAD( INTERPOLATE(A,Nb',γ),γ,range)  ]
```

Given the axioms above, it is clear that we can write the following propositions or theroems for the first two map overlay operations.

```
ADD( A,B) = ADD( B,A )
ADD( A, ADD( B,C )) = ADD( ADD( A,B ), C)

¬ COVER( A,B) = COVER( B,A )
COVER( A, COVER( B,C )) = COVER( COVER( A,B ), C)
```

For example, we can write the definitions for isotims and isodapanes using ADD and SPREAD. Granted the example is a bit contrived, but it serves its purposes nonetheless. The isotims between 0 and r, from a location, g, on a weighted terrain, m_0, are computed from SPREAD(m_0,g,r). If another map, m_1, contains the terminal costs, then the isodapanes for the complete transportation problem are represented on a map as:

$$M = ADD(SPREAD(m_0,g,r), m_1).$$

Tomlin's Map Algebra, and also the specification we have written, are essentially homogeneous algebras. However, these are not particularly interesting in the sense that we cannot do much with it. Because the algebra is homogeneous, the language does not support queries such as the comparison of two maps. In the sections below, we shall explore a specification for prescriptive spatial modelling.

4. Objected-oriented Programming Language

Thus far we have not quite touched on the subject of object oriented programming language. We described the map overlay methodology in environmental planning, we reviewed some of its history as a manual technique and presented a computer-assisted planning tool as a map overlay metaphor. In the last section we described a specification for Map Algebra. Given a specification as presented in appendix A, this section briefly describes an implementation of the specification in *C++*. An example is given. In the final section we contrast this object oriented implementation of Map Algebra with that of the original.

4.1. Concepts in Object Oriented Programming

The basic notion of object oriented programming, perhaps being a tautology in definition, is that everything in the system is an object.

It is difficult to find an agreed upon definition for object oriented programming language. Data hiding, data abstraction, inheritance or subclass, communicating objects, and modularity are features commonly associated with object oriented programming. There are now many different language implementations of object oriented programming and they vary tremendously. Since most of the current object oriented programming languages have some lineage to ***SIMULA***, the following definitions, found in the ***SIMULA*** language, should be of use:

> (i) An *object* in general consists of three parts: a heading identifying the kind of object, a data structure and an action sequence. (ii) A group of objects having the same heading, layout of variables and action pattern are said to belong to the same *class* (Birtwistle, 1973, 26).

Nevertheless, the most important aspect in object oriented programming is really not its technical advantages: i.e., the language enforcement of inheritance or data hiding, it is the fact that object oriented programming allows the programmer to design programs in a more natural manner. When a problem is divided into objects, these objects carry with them certain tightly related information and a set of actions. An algorithm that is described in terms of objects is much easier to develop because the programmer needs only to consider details and specifics found within the particular object he is working on and not the details and specifics from the rest of the system. The programmer can feel safe to concentrate on a particular object because he is assured that the object is accessible to the outside world only through the message interface, a set of public functions defined for the object.

4.2. Map Algebra Examples

Based upon the map overlay metaphor, it is straightforward to define the overlays described above as object classes. Hence, each map overlay has a private data structure: the dimensions of the matrix, the data matrix itself and the name of the overlay.

```
class map_overlay {
        int          nRow,nCol;
        double     mapScale;
        double     mapX, mapY;
        char *       mapNote;
        double *   leemetz;
};
```

All of these data are, of course, private to the object; they are hidden from the user of the object. Object are accessible only through their public member functions. For example, we have two such functions, Point and getGPoint, which sets the value at a grid point location on an overlay and retrieves a value, respectively.

```
class map_overlay {
        // ...
public:
        void    Point(gPoint g, double value);
        double  getGPoint(gPoint g);
};
```

The following piece of code illustrates the use of class map overlay and its, four at the moment, set of operations. The constructor and destructor take care of all the house keeping operations in creating and cleaning up of dynamic storage of object in class map overlay. Class map overlay defines a type in the language, a data object, *overlay*, must first be declared, and it is constructed automatically. The dot operator in the subsequent statements selects the class member function that is sent to the newly created object *overlay*.

```
struct gPoint { int col, row; }

class map_overlay {
        // ...
public:
        map_overlay();                          // constructor
        ~map_overlay();                         // destructor
        // ...
};

main()
{       double value;
        gPoint g = { 1,1 };
        map_overlay overlay;

        overlay.Point( g, 99.0);
        value = overlay.getGPoint( g);
}
```

Recall from the earlier discussions of Map Algebra, we have three classes of operators: point, neighbourhood and region. In the conventional programming language, we probably would have a common data structure, something like that found in the private part of our class map overlay. There is no way, except with some tedious conditional statements inside each function, to tell the program whether the correct operation is applied to the suitable data map; and, more importantly, there is no way to prevent the user from doing something unexpected; for example, in the Map Analysis Package, doing a regional size on an essentially continuous three-dimensional surface map layer or doing a neighbourhood profile on zoned thematic map layer. Sizeing a continuous surface obviously returns the grid cell dimension, faithfully. Profileing a zoned map is actually rather fun because it is a roundabout way to do edge detection, only all edges are known on a zoned map, by definition. To better organize our thoughts in problem solving and to provide a much needed flexibility for future developments of the computerized map overlay methodology, these different classes of operations be represented as different object classes. And, in object oriented programming, this can be done painlessly. If we consider the base map layer class as a planimetric map, then the class with added relief or neighbourhood operations is a topographic map and the one with region operations is thematic.

```
class topographic_layer:                    // terrain ops.
   friend map_layer {
         double *   elevation;
public:
         profile();
         // ...
};

class thematic_layer:                       // region ops.
   friend map_layer {
         region   *regionList;
         clump    *clumpList;
         // ...
public:
         size();
         // ...
};
```

The topographic layer and the thematic layer inherit all the component parts from the base map layer class. This example of inheritance demonstrates the ability to reuse some very much standard code in object oriented programming languages. For instances, most object oriented programming languages have predefined classess for generic link-list, tree, etc. If one wants to implement some of McHarg's cost benefit analysis functions into the basic package, he can simply add, perhaps, an 'utility' layer with special member functions and data fields but retains the existing input and display functions. He can also override some of the existing spatial and arithmetic operations with special ones.

There is also another illustration of modularity in object oriented programming shown in the thematic layer class definition. We have attached two classes: region and clump inside thematic layer; for all practical purposes, there is no requirement at all that these two classes should have a similar grid cell structure as found in the map layer class, their superclass. They are entirely hidden from the user but we can implement any type of geographic information processing operations on them as we wish.

4.2.1. Research Agenda

Stroustrup describes the aim of the *C++* class concept "to provide the programmer with a tool for creating new types that can be used as conveniently as the built-in types. Ideally, a user-defined type should not differ from built-in types in the way it is used, only in the way it is created" (Stroustrup, 1986, 133). This is easily demonstrated with the simple point characterizing functions. Let consider the AVERAGE function in the original Map Analysis Package. The syntax of that functions is:

AVERAGE overlay$_1$ TIMES weight$_1$ [PLUS overlay$_n$ TIMES weight$_n$] FOR overlay

When we treat the user-defined overlay type as if it were a built-in type, the AVERAGE function becomes regular program code:

```
main()
{
        map_overlay A,B,C;
        double wgt;

                        //      AVERAGE a TIMES wgt
                        //        PLUS b  FOR C

        C = (( A * wgt) + B ) / 2;
}
```

This example might seem trivial; perhaps, it is. But, consider the flexibility this allows in the construction of some basic statistics calculations, say, variance over a number of map overlays. The truly exciting aspects of the object oriented paradigm are the added flexibility found in the manipulations of the other two classes of operations: neighbourhood and region. Research on those operations are on-going.

5. Concluding Remarks

This paper reported an object oriented implementation of Map Algebra. From his own Map Analysis Package, Tomlin, in 1983, derived an algebra to formalize the "map overlay" planning methodology.

The increased sophistication apparent in today's geographic information system users warrants our efforts in this research. Our goal is to implement a well-defined Map Algebra for application specialists. Furthermore, since our implementation is based on a formal algebra, it is a particularly significant experiment for future construction of geographic expert systems and in providing a means for the specification of geographical rules for land planning. Formal algebra often times raises concerns about implementation difficulties. We demonstrated that it is possible to easily translate an algebra into a general purpose programming language using object oriented programming.

6. Bibliography and References

Association for Planning and Regional Reconstruction (1950).*Town and Country Planning Textbook*. The Architectural Press, London.

Birtwistle, GM, O-J Dahl, B Myhrhang and K Nygaard (1973). *SIMULA BEGIN*. Auerbach Publishers Inc., Philadelphia.

Burrough, P (1986). *Principles of Geographic Information Systems for Land Resources Assessment*. Oxford University Press, New York.

Chrisman, NR (1979). "Concepts of space as a guide to cartographic data structures," *Harvard Papers on Geographic Information Systems*, Geoffrey Dutton, Editor, Volume Seven. Harvard Laboratory for Computer Graphics and Spatial Analysis, Cambridge, Massachusetts.

_________ (1987). "Fundamental principles of geographic information systems," *Auto-Carto 8*, pp. 32-41.

Cox, BJ (1986). *Objective Oriented Programming: An Evolutionary Approach*. Addison-Wesley, Massachusetts.

Goldberg, A and D Robson (1983). *Smalltalk-80: The Language and its Implementation*. Addison-Wesley, Massachusetts.

Guttag, J V and J J Horning (1978). "The algebraic specification of abstract data types," *ACTA Information*, Volume 10, 1978, pp. 27-52.

Herring, D (1987). "TIGRIS: topologically integrated geographic information system," *Auto-Carto 8*, pp. 282-291.

Manola, F and JA Orenstein (1986). "Toward a general spatial data model for an object-oriented DBMS," *Proceedings of the Twelfth International Conference on Very Large Data Bases*, pp. 328-335.

Manola, F, JA Orenstein and U Dayal (1987). "Geographic information processing in the PROBE database system," *Auto-Carto 8*, pp.316-325.

Martin, J J (1986). *Data Types and Data Structures*. Series in Computer Science, Prentice-Hall International, Eglewood Cliffs, New Jersey.

Morehouse, S and M Broekhuysen, (1982). *Odyssey User's Manual*, Harvard Laboratory for Computer Graphics and Spatial Analysis, Cambridge, Massachusetts.

Nygaard, K (1986). "Basic concepts in object oriented programming," *SIGPLAN* Notices, Volume 21, No. 10, October, pp. 128-132.

Sinton, DF (1978). "The inherent structure of information as a constraint to analysis: mapped thematic data as a case study," *Harvard Papers on Geographic Information Systems*, Geoffrey Dutton, Editor, Volume Seven. Harvard Laboratory for Computer Graphics and Spatial Analysis, Cambridge, Massachusetts.

_________ (1979). "An introduction to I.M.G.R.I.D., An information manipulation systems for grid cell data structures," *Mapping Software and Cartographic Data Bases*, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, pp. 180-194.

Steinitz, C, P. Parker and L Jorden (1976). "Hand-drawn overlays: their history and prospective uses," ***Landscape Architecture***, volume 66, No. 8, pp. 444-455.

Stroustrup, B (1986). ***C++ Programming Language***. Addison-Wesley, Massachusetts.

Tomlin, CD (1983a). Digital Cartographic Modeling Techniques in Environmental Planning. Doctoral dissertation. Division of Forestry and Environmental Studies, Yale Graduate School.

__________ (1983b). "A Map Algebra," ***Proceedings of the 1983 Harvard Computer Graphics Conference***, Volume II. Harvard Laboratory for Computer Graphics and Spatial Analysis, Cambridge, Massachusetts.

__________ (1986). The IBM Personal Computer version of the Map Analysis Package. GSD/IBM AcIS Project, Report No. LCGSA-85-16. Harvard Laboratory for Computer Graphics and Spatial Analysis, Cambridge, Massachusetts.

Tyrwhitt, J (1950). "Surveys in planning," in ***Town and Country Planning Textbook***, Edited by the Association for Planning and Regional Reconstruction. The Architectural Press, pp. 146-178.

Formal Specifications of Map Analysis Package (MAP)

Introduction

A specification for MAP is presented below. There are slight variances between the followings and the descriptions of the language in the MAP manual (Tomlin, 1986); however, most of the differences are found in the treatment of optional arguments, and are insignificant for the interpretation of the algebra. The functional description of the language presented in the manual is still valid, with the exception of the command "map." In our specification, the "map" command creates only one map layer. All pointwise arithmetic operations are lumped together in the ⊗ operator in the base class "map layer." Display and data management operations are not specified.

MAP commands that are not defined in this specification are: average, describe, display, echo, explain, expose, grid, label, list, note, print, protect, quiet, read, show, strip, trace, write and zap. Grid, read, strip and trace can be easily defined by the "point" operation in the base class "map layer." Whereas, describe, display, list, print, show and write are data output commands; they can be found in numerous graphics standards specifications. Average has been superseded. Echo, expose, protect, quiet and zap are mainly programming control commands; explain, label and note are programming utilities. They are, hence, deleted from the definition of the algebra.

The following section briefly defines some of the terms used in the specification. The specification is itself separated into three parts. Part (i) presents the base class "map layer." Part (ii) is the subclass "topographic layer." "Thematic layer" is found in part (iii).

Definitions:

A "map layer" is the most elementary data element in MAP; it contains the map image, information that defines the "dimensions" of the layer, and other documentary items. The map image is composed of locational points; points defined on the image are of type "gPoint." Groups of points are "pSets." "Dimensions" of a map layer determine the global geographic reference, the geographic extents, and the map-to-ground scale for the map layer. By and large, we can consider the basic map layer as a planimetric map of the study area.

A "topographic layer" is a subclass of the map layer class that, in addition to all the planimetric characteristics of a basic map layer, represents and manipulates the study area as if it is a continuous surface. "Neighbourhood" is defined on topographic layer as piecewise continuous surface element of the surface which covers the entire study area.

A "thematic layer" is another subclass of the map layer class. The study area is treated as a discrete surface composed of a finite number of "clumps." "Clumps" are groups of points that share a common "value" on the layer from which they are constructed, furthermore, points in a clump are topologically connected and the diameter of any clump must be within the "range" specified in the thematic layer generator function. "Regions" are groups of "clumps" with the same value but which may or may not be connected.

"gPoint" and "pSet" can be associated with "values." A "value" can be of any arbitrary data type. "vRange" delineates a selection criterion over "value." A "gRange" is a distance measurement that is employed in the "topographic layer" class to map out a "neighbourhood" for a "gPoint," or to define a "clump" in the "thematic layer" class.

Finally, "gPoints" are identified through the geographic reference system. All "gPoints" on a topographic layer must have relief information. "Regions" and "clumps" on a thematic layer must have unique identification numbers.

Type: **map_layer**

Set: map_layer, gPoint, dimension, value, vRange

Syntax:

⊗	map × map	→ map
assign	map × value	→ map
copy	map	→ map
cover	map × map	→ map
cross	map × map × value × vRange	→ map
maximize	map × map	→ map
map	dimensions	→ map
minimize	map × map	→ map
point	map × gPoint × value	→ map
slice	map × NumOfInterval × vRange	→ map

axioms:

⊗(map(dem), m) = error   where, ⊗ = { add, divide, exponentiate, multiply, subtract }

⊗(m, map(dem)) = error

⊗(point(m_1,g,v_1), point(m_2,g,v_2)) = point(m,g, $v_1 \otimes v_2$)

assign(map(dem), v) = point(m,g,v)

assign(point(m,g,v_1), v) = point(m,g,v)

copy(map(dem)) = error

copy(gPoint(m,g,v)) = gPoint(m,g,v)

cover(map(dem),m) = error

cover(m, map(dem)) = copy(m)

cover(point(m_1,g,v_1), point(m_2,g,v_2))

    = if ($v_1 \neq 0$) then point(m,g,v_1) else point(m,g,v_2)

cross(map(dem),m,v,vRange) = error

cross(m, map(dem),v,vRange) = error

cross(point(m_1,g,v_1),point(m_2,g,v_2),value,vRange)

    = if (($v_1 \in$ vRange) *and* ($v_2 \in$ vRange)) then point(m,g,value) else point(m,g,0)

maximize(map(dem), m) = error

maximize(m, map(dem)) = error

maximize(point(m_1,g,v_1),point(m_2,g,v_2))

    = if ($v_1 \geq v_2$) then point(m,g,v_1) else point(m,g,v_2)

minimize(map(dem), m) = error

minimize(m, map(dem)) = error

minimize(point(m_1,g,v_1),point(m_2,g,v_2))

    = if ($v_1 \leq v_2$) then point(m,g,v_1) else point(m,g,v_2)

slice(map(dem), i, vRange) = error

slice(point(m,g,v), i, vRange)

    = if ($v \in$ vRange) then point(m,g, classify(v,vRange,i))

Type: **topographic_layer**

Set: topographic_layer, neighbourhood, map_layer, gPoint, gRange, value

Syntax:

diff	topographic_layer	→ map
diffMax	topographic_layer	→ map
drain	topographic_layer × map	→ map
metric	map × vScale × hScale	→ topographic_layer
orient	topographic_layer	→ map
orientPrec	topographic_layer	→ map
profile	topographic_layer × bearing × ε	→ map
profileInflec	topographic_layer × bearing	→ map
radiate	topographic × gPoint × gRange	→ map
scan	topographic × statFun × map × gRange	→ map
scanDiag	topographic × statFun × map × gRange	→ map
spread	topographic × gPoint × gRange	→ map
spreadDown	topographic × gPoint × gRange	→ map
spreadWeight	topographic × gPoint × gRange × map	→ map

hidden functions:

interpolate	topographic × neighbourhood × gPoint	→ topographic_layer

axioms:

g, γ ∈ { geographic point }, and Nb ∈ { neighbourhood }

implicit function: Φ(metric(m,v,h)) = ∀(g)(g∈m) Φ(interpolate(m,Nb,g))

```
diff( interpolate(M,Nb,g)) = point( m,g, SumOfSlope(Nb))
diffMax( interpolate(M,Nb,g)) = point( m,g, MaxOfSlope(Nb))
drain( interpolate(M,Nb,g), planiMap) = ∀(γ)(γ ∈ Nb)
        if ( g.elev ≥ γ.elev ) then
           point(m,g, planiMap.value += drain( interpolate(M,Nb,γ),planiMap).γ )
```

```
orient( interpolate(M,Nb,g))
        = point( m,g, aspectType(Nb))
orientPrec( interpolate(M,Nb,g))
        = point( m,g, aspect(Nb))
```

```
aspectType = {
   non-sloping areas;
   slopes facing the top edge of a map;
   slopes facing the upper right corner of a maps;
   slopes facing the right edge of a map;
   slopes facing the lower right corner of a map;
   slopes facing the bottom edge of a map;
   slopes facing the lower left corner of a map;
   slopes facing the left edge of a map;
   slopes facing the upper left edge of a map;
}
```

```
profile( interpolate(M,Nb,g), bearing,ε))
     = point(m,g,crossSectionType(Nb,bearing,ε))
profileInflex( interpolate(M,Nb,g),bearing))
     = point(m,g,exteriorAngle(Nb,bearing))
```

```
crossSectionType = {
   profiles that extend "up and up,"
   profiles that extend "up and over,"
   profiles that extend "up and down,"
   profiles that extend "over and up,"
   profiles that extend "over and over,"
   profiles that extend "over and down,"
   profiles that extend "down and up,"
```

```
        profiles that extend "down and over,"
        profiles that extend "over and over"
        profiles that extend "over and over"
    }
```

where,

"up" means > ε, "down" means < ε, and "over" means ≤ | ε |.

bearing = {north, northeast, east, southeast, south, southwest, west, northwest }

```
radiate( interpolate(M,Nb,g),g,range) = ∀(γ)(γ ∈ Nb)
      if ( visible(g,γ) and (dist(g,γ) ≤ range))
        then point(m,g,true)   else point(m,g,false))
scan( interpolate(M,Nb,g), statFun,range)
     = G = { γ | (RookDist(g,γ)≤range)}
        point( m,g, Fun(kMap,G) )
scanDiag( interpolate(M,Nb,g), statFun,range)
     = G = { γ | (QueenDist(g,γ)≤range)}
        point( m,g, Fun(kMap,G) )
spread( interpolate(M,Nb,g),g,range)
     = ∀(γ) (γ ∈ Nb)
        [ cost = g.value+distance(γ,g)
        if ( (cost ≤ range) and  ((γ.value>cost)or(γ.value = null)) )
          point(m,γ, g.value+distance(γ,g)
          spread( interpolate(M,Nb',γ),γ,range) ]
spreadDown( interpolate(M,Nb,g),g,range)
     = ∀(γ) (γ ∈ Nb)
        [ cost = g.value+distance(γ,g)
        if ( ((cost ≤ range) and (γ.elev ≤ g.elev)) and ((γ.value>cost)or(γ.value = null)) )
          point(m,γ, g.value+distance(γ,g)
          spread( interpolate(M,Nb',γ),γ,range) ]
spreadWeight( interpolate(M,Nb,g),g,range,point(wMap,g,w) )
     = ∀(γ) (γ ∈ Nb)
        [ cost = g.value+ (distance(γ,g) · w)
        if ( (cost ≤ range) and  ((γ.value>cost)or(γ.value = null)) )
            point(m,γ, g.value+distance(γ,g)
            spread( interpolate(M,Nb',γ),γ,range) ]
```

Type: **thematic_layer**

Set: thematic_layer, map_layer, gRange, pSet, value

Syntax:

clump	map × gRange	→ thematic_layer
clumpDiag	map × gRange	→ thematic_layer
euler	thematic_layer	→ thematic_layer
recode	thematic_layer × pSet × value	→ thematic_layer
score	thematic × map × statFun	→ thematic_layer
size	thematic_layer	→ thematic_layer
sort	thematic_layer	→ thematic_layer
span	thematic_layer × gPoint × gRange	→ map
survey	thematic_layer	→ thematic_layer

hidden functions:

addpSet   thematic_layer × pSet × value   → thematic_layer

Axioms:

```
clump( map(dem), range) = error
clump( point(m,g',v), range)
        = G = { g | (connect(g',g) and (RookDist(g,g') ≤ range)) }
          addpSet(theme,G, v)
clumpDiag( map(dem), range) = error
clumpDiag( point(m,g',v), range)
        = G = { g | (connect(g',g) and (QueenDist(g,g') ≤ range)) }
          addpSet(theme,G, v)
euler(addpSet(theme,G,v)) = recode( theme, G, NumOfEdges(pset))
recode( addpSet( theme, G, v), G, value) = addpSet( theme, G, value)
score( theme, map(dem), function)
          = error
score( addpSet(theme,G,v), m, function)
          = M = { m | ( m.g ∈ G)) }
            recode(theme,G, function(M.v))
size( addpSet(theme,G,v)) = recode( theme, G, area(G))
sort( addpSet( addpSet( theme, G', v'), G, v))
        = if (v < v') then recode( theme,G, id)  recode( theme, G', id'), id < id'
span( addpSet(theme,G,v), g,range) = point(m,g, diamAtPoint(G,g))
survey( addpSet(theme,G,v)) = recode( theme, G, shape(G))
```

where, function = { total, average,
maximum, minimum,
median, majority, diversity,
deviation and proportion }

Heading file for class map layer.

```
/*********** ccMap.h ****************/

#include <string.h>
struct gPoint { int rIndex, cIndex };
class map_layer;
overload pow;
extern double pow(double, double);
map_layer pow(map_layer,map_layer);
map_layer pow(map_layer, double);
overload ceil;
extern double ceil(double);
map_layer ceil(map_layer);
overload floor;
extern double floor(double);
map_layer floor(map_layer);
overload map_layer maximize(map_layer&,map_layer&), maximize(map_layer&,double);
overload map_layer minimize(map_layer&,map_layer&), minimize(map_layer&,double);
overload map_layer cross(double,map_layer&,double,double,map_layer&,double,double),
        cross(double, map_layer&,double,map_layer&,double);
overload map_layer recode(map_layer&,double,double,double), recode(map_layer&,double,double);

/*
 *      class map_layer is the basic class that handles all point-wise arithmetic
 *       and relational operations.
 */

class map_layer {
        int nRow, nCol;
        double mapScale, xOrigin, yOrigin;
        char *mapNote;
        double *image;                                                  // grid matrix;
public:
        map_layer( int r, int c)                                        // constructor
                { nRow = r; nCol = c; image = new double [ r*c ]; }
        map_layer(map_layer&);
        ~map_layer()            { delete image; }                       // destructor
        void load(char*);                                               // read a map from file
        void print();
        void jot(char* co)
                { delete mapNote; mapNote = new char[ strlen(co)+1 ];
                  strcpy(mapNote,co);     }
        void reshape(int,int,double,double);
        void scale(double);
// Map Analysis Package (extended point-wise oeperations)
        friend map_layer operator+(map_layer&,map_layer&);
        friend map_layer operator+(map_layer&,double);
        void operator+=(double);
        friend map_layer operator&&(map_layer&,map_layer&);                     // AND
        void operator=(double);                                                 // assign
        void ceil();
        friend map_layer ceil(map_layer&);
        void operator=(map_layer&);                                             // copy
        friend map_layer cover(map_layer&,map_layer&);
        friend map_layer cross(double,map_layer&,double,double,map_layer&,double,double);
        friend map_layer cross(double,map_layer&,double,map_layer&,double);
        char* describe()    {return (mapNote); }
        friend map_layer operator/(map_layer&,map_layer&);
        friend map_layer operator/(map_layer&,double);
        void operator/=(double);
        friend map_layer operator==(map_layer&,map_layer&);
        friend map_layer pow(map_layer&,map_layer&);
        friend map_layer pow(map_layer&,double);
        void pow(double);
        friend map_layer floor(map_layer&);
        void floor();
        double getPoint(gPoint);
        friend map_layer operator>(map_layer&,map_layer&);
        friend map_layer operator>=(map_layer&,map_layer&);
        friend map_layer operator<(map_layer&,map_layer&);
        friend map_layer operator<=(map_layer&,map_layer&);
        friend map_layer maximize(map_layer&,map_layer&);
        friend map_layer maximize(map_layer&,double);
        friend map_layer minimize(map_layer&,map_layer&);
        friend map_layer minimize(map_layer&,double);
        friend map_layer operator*(map_layer&,map_layer&);
        friend map_layer operator*(map_layer&,double);
        void operator*=(double);
        friend map_layer operator||(map_layer&,map_layer&);                    // OR
        void point(map_layer&,gPoint,double);                                  // set point
        friend map_layer operator-(map_layer&,map_layer&);
        friend map_layer operator-(map_layer&,double);
        void operator-=(double);
};
```

SOME TOPOGRAPHIC INGREDIENTS OF A GEOGRAPHIC INFORMATION SYSTEM

Richard J. Pike[1], William Acevedo[2], and Gail P. Thelin[2]

U.S. Geological Survey, [1]Menlo Park, CA 94025 and
[2]Ames Research Center, Moffett Field CA 94040

ABSTRACT

Topography is a basic component of any geographic information system (GIS). Two computer techniques, digital image processing and the triangulated irregular network (TIN), extract topographic measures useful for GIS analysis from large matrices of altitudes. Image processing rapidly calculates such measures from altitude rasters and displays them as maps. The TIN organizes terrain into surface-specific facets through a raster-to-vector transformation. Both methods have been applied to a new 30-m-resolution digital elevation model (DEM) in San Mateo County, California, which is free of the systematic error that degrades older DEMs. The resulting raster-image maps and TINs of slope, slope reversal, aspect, curvature, and hypsometry are potential ingredients of a quantitative topographic base for a GIS. Syntheses of these and other variables can yield maps for planning, environmental studies, and geomorphic analysis.

1. INTRODUCTION

Large sets of topographic data and advanced ways of analyzing them are adding important capabilites to geographic information systems (GIS). While mass-production of altitude matrices has all but replaced the laborious data-capture that once so restricted the quantitative description of land form (1, 2), new computer methods are automating the characterization of topography (3-5). GIS-based applications of these developments in terrain analysis include geomorphic interpretation as well as contributions of land-surface form to spatial models for planning, assessing environmental hazards, land-type classification, and risk analysis (2, 4, 6, 7).

Contour maps and their digitized altitudes are only the most common topographic ingredients of a GIS. We believe that derivatives of altitude can better represent terrain for spatial correlation, classification, and interpretation. The resulting topographic base (7) should elaborate the original altitude matrix and comprise several mappable variables (e.g., slope) that are compatible with other GIS coverages (1, 4, 5, 7, 8). Unit cells for such maps may be small watersheds, slope facets, or single altitudes.

This paper describes recent computer experiments in the numerical characterization of 1:24,000-scale terrain by altitude and its derivatives (7). The immediate goal is a strong topographic component in the GIS currently being developed for San Mateo County,

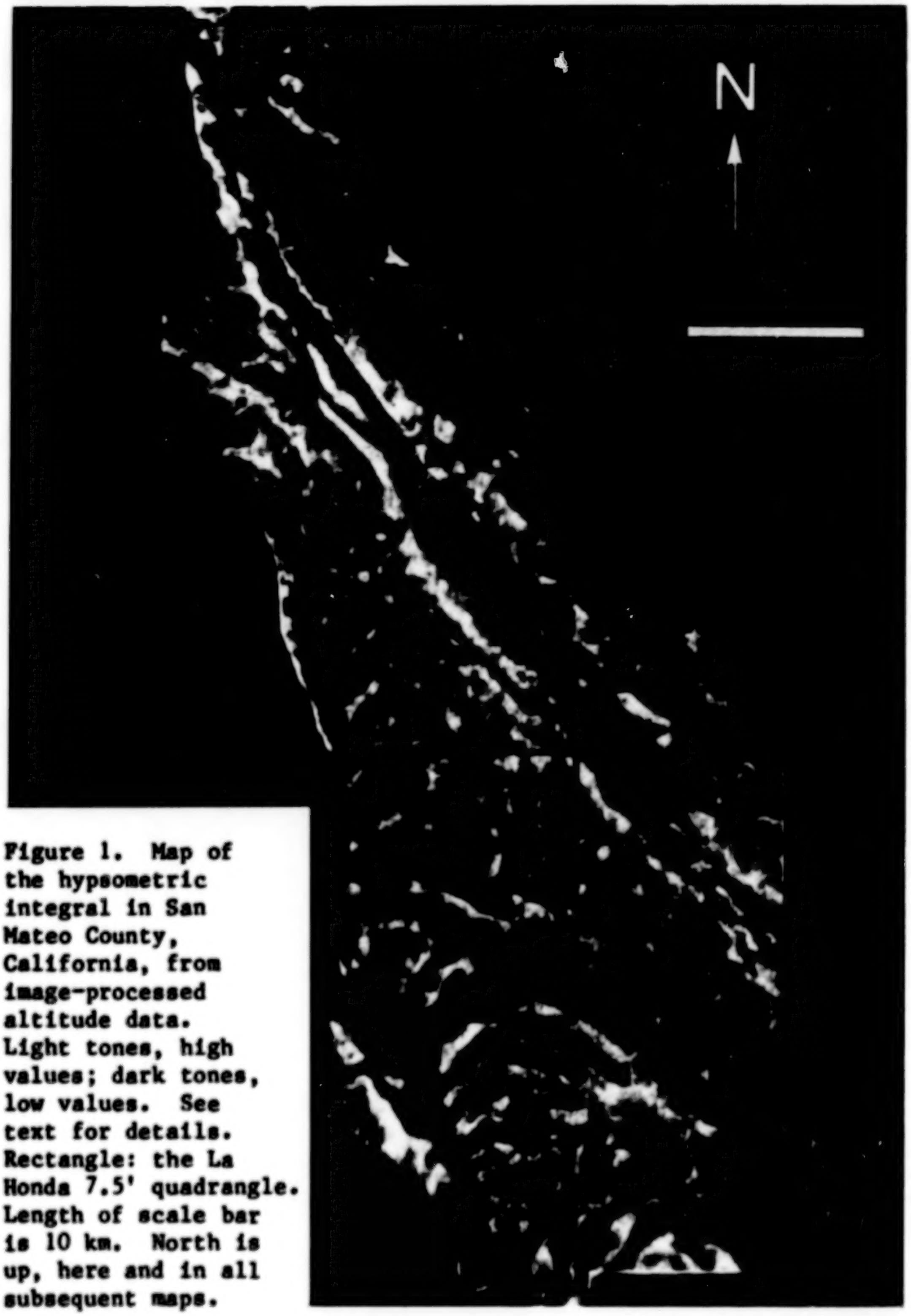


Figure 1. Map of the hypsometric integral in San Mateo County, California, from image-processed altitude data. Light tones, high values; dark tones, low values. See text for details. Rectangle: the La Honda 7.5' quadrangle. Length of scale bar is 10 km. North is up, here and in all subsequent maps.

California (2; fig. 1), but the concepts pertain to GIS analysis generally. Current products of our pilot study, which uses the La Honda 7.5' digital elevation model (DEM), are maps from both digital image processing and triangulated irregular network (TIN) techniques. This paper complements and expands our initial results (7) by presenting new maps, mainly from image processing.

2. THE LA HONDA SAMPLE

The DEMs distributed by USGS for 7.5' quadrangles in the lower 48 states are prone to systematic error (2, 7). Errors in altitudes

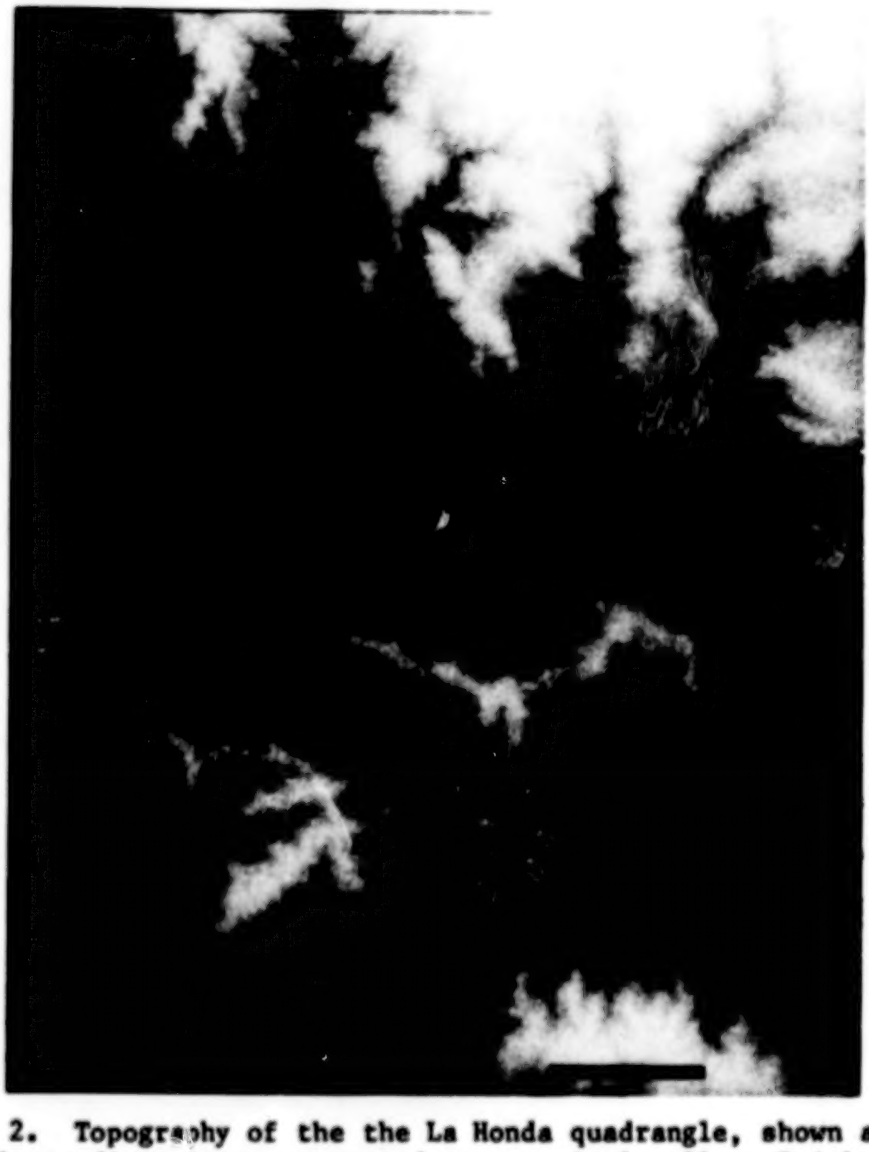


Figure 2. Topography of the the La Honda quadrangle, shown as an altitude-gradient image processed on one-pixel cells. Bright tones, high terrain; dark tones, low terrain. Length of scale bar is 2 km.

for adjacent rows or columns appear as stripes aligned in the direction of compilation. No such error is evident in the revised DEM for La Honda (figs. 8, 9 of ref. 2), the first of 15 quadrangles in San Mateo County to be compiled by digitizing and gridding the original contour plates rather than by scanning a stereomodel. This improved DEM, measuring 465 x 372 altitudes, is the first to have the accuracy required for detailed analysis of 30-m topography.

The La Honda quadrangle lies on the west slope of the Santa Cruz Mountains in the California Coast Ranges (fig. 1). Tertiary sedimentary and volcanic rocks (6) are intricately carved into steep ridges and valleys (fig. 2). Main features include streams with narrow flood-plains, minor control of terrain by faults, and many landslides. The quadrangle has 640 m of local relief, mean altitude is 209 m, mean slope is about 17°, ridges and valleys average about 210 m apart, and mean ridge-to-valley depth is 45 m (7).

3. RASTER IMAGE PROCESSING

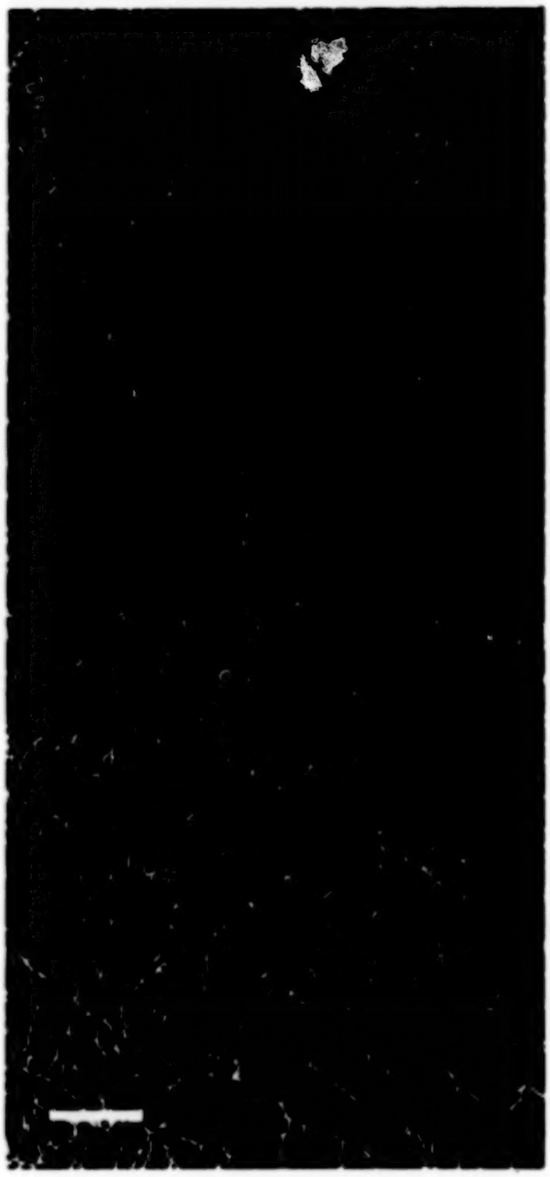
First developed for reassembling pictures from spacecraft telemetry, digital image processing was adapted to mapping altitude matrices and their derivatives over large areas (9). It is a more mature technology than TIN (3). The raster data structure, one pixel = one altitude, enables the quantitative measures that describe terrain to be rapidly calculated, compared, and combined for display as GIS-based map images in color or shades of grey. Prior applications of image processing to topography have tended to be at regional or broader scales, and only for altitude, slope, or aspect (7, 9, 10). The images presented here (figs. 1, 3-6) show more complex derivatives of comparatively local, 1:24,000-scale, terrain (7).

Most digital-image maps were generated on a Sun* workstation using software still under development. Some maps were compiled on a VAX 11/780* computer by the Interactive Digital Image Manipulation System (IDIMS; 11). The new images are of percentage hypsometry, slope reversal and its frequency, slope curvature and its mean (figs 1, 3-6), and relative relief (not shown). Many of the maps were processed by calculating values of each variable within windows -- typically 13x13, 25x25, and 49x49 pixels -- moved one to five pixels at a time through the entire DEM. Generally, the images shown here are continuous-tone portrayals (255 grey levels) of the results.

Figure 1, a map of the hypsometric integral for La Honda and its surrounding quadrangles, depicts localized altitude-distribution of terrain. The parameter is commonly applied to hydrologic problems within single watersheds, but only rarely is used as a descriptor of continuous topography (1, 8, 12), where, as here, it complements other variables. Figure 1 was generated for all of San Mateo County (from the older DEMs) using a simple (and identical) equivalent, the elevation-relief ratio: ER = [mean - minimum elevation / relief] (12), on a 25x25 pixel window moved every five pixels.

---

*Trade names and trademarks in this paper are for descriptive purposes only, and imply no endorsement by the U.S. Geological Survey.



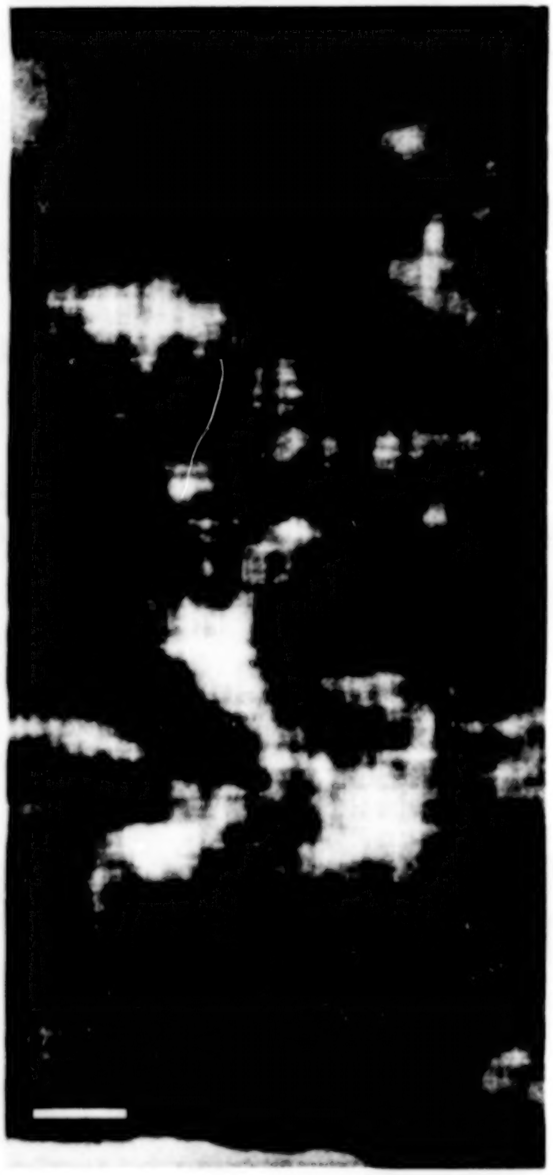
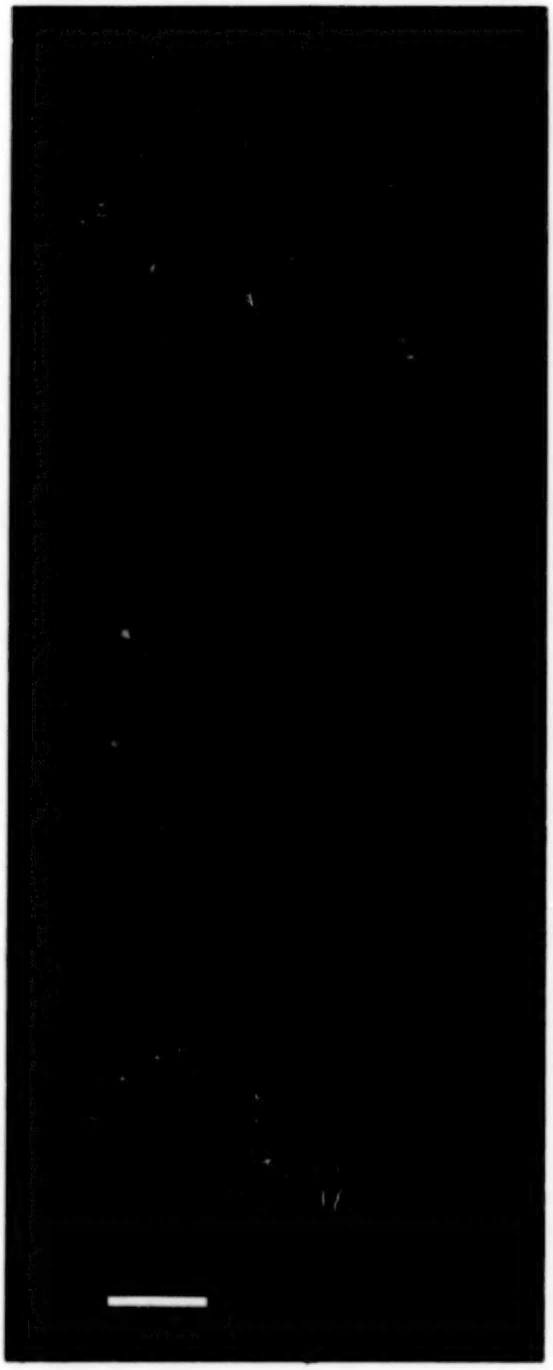


Figure 3. (left) Drainageway (heavy lines) and ridgeline (light lines) net, in central half of the La Honda quadrangle, derived from reversals of slope in the DEM (see text). Scale bar is 1 km long.

Figure 4. (right) Frequency of slope reversal in the central La Honda quadrangle, mapped by image processing of figure 3. Bright areas are fine textured topography; dark areas are coarse textured.



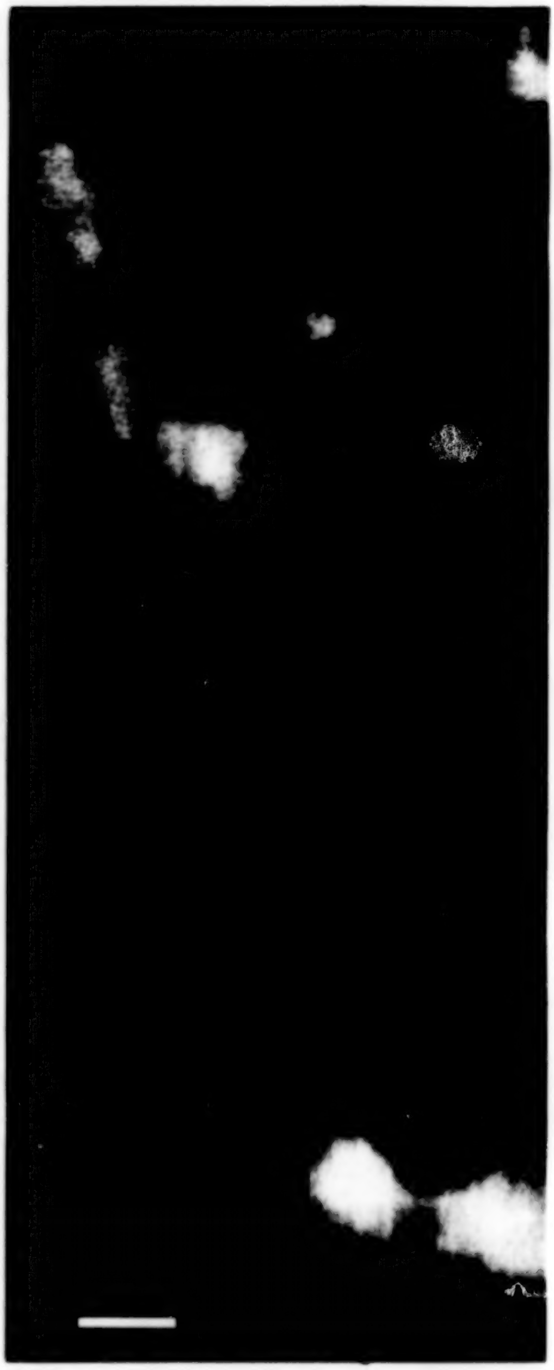


Figure 5. (left) Topographic curvature measured at E-W reversals of slope in the central half of the La Honda quadrangle. Brightest lines indicate sharpest valleys and ridges. Scale bar is 1 km long.

Figure 6. (right) Mean angle of curvature-between-reversals, in the central La Honda area, mapped continuously by image processing of the data in figure 5. Bright areas are roughest terrain.

ER measures the percentage distribution of volume with altitude; in homogeneous eroded terrain of at least moderate relief, it reflects the relative proportion of unconsumed upland. Values of ER lie between 0.0 and 1.0. High values (light areas in fig. 1) are upland surfaces incised by few drainageways, and low values (dark areas) are lowlands with few erosional outliers of upland. Intermediate values can indicate uneven terrain with fairly regular slopes that may be in an equilibrium or steady-state condition. Values of ER in low-relief terrain (e.g. upper right of figure 1, along San Francisco Bay), probably have little geomorphic significance.

Texture, the spacing of landforms in plan, is as important as slope and percentage hypsometry for characterizing terrain (1, 4, 8). Texture in the La Honda area is expressed here by slope direction change (SDC), the reversal of slope along a DEM rank or file. By convention, positive slopes face the start of a profile. Figure 3 shows that negative-to-positive reversals map much of the drainage net (bold lines), while positive-to-negative reversals outline the ridge systems (thin lines), albeit discontinuously (cf. figs. 9, 10). In figure 4, this discrete line network is converted to the frequency of SDC (within a 25 x 25 moving window), a continuously mappable quantity. Light tones indicate high-frequency spacing, or fine texture, the dark tones coarse texture. This image of the La Honda area is unlike maps of slope and related variables because texture differs substantially from those other properties. Both figures 3 and 4 contain statistical noise, which can be reduced by removing SDCs that measure low-angle reversals of slope.

Terrain roughness may be measured independently of steepness, or slope angle (8). To this end we use curvature, the first derivative of slope (4), or relative slope. It is defined here as the supplement of an angle subtended by three adjacent altitudes along a DEM rank or file. Slope curvature for the La Honda area is mapped in figure 5 explicitly at slope reversals, and only those found along E-W DEM ranks in figure 3. The brighter the line segment in figure 5, the larger the angle of curvature and the rougher the terrain. Figure 6 maps the corresponding distribution of average curvature, computed on a 25 x 25 pixel moving window. Bright areas indicate the steepest terrain. Figure 6 mimics maps of altitude, relative relief, and slope angle (7) in the La Honda quadrangle.

Parameter values calculated from a DEM by image-processing at each pixel or window position can be variously tabulated, summarized, and analyzed, as in Table 1. Figure 7 shows that mean values of slope curvature are a positive linear function of slope angle (image not shown) for the 11 most common geologic units mapped in the La Honda quadrangle (6, fig. 1). Also, according to mean values of slope curvature (Table 1), terrain texture (SDC) becomes somewhat finer with increasing slope and curvature. Finally, six map units ($S_{1,2}$, $C_{1,2}$, $J_{1,2}$) paired with respect to engineering properties (6, Table 1) also tend to pair up, however roughly, according to all three

Table 1. Mean Terrain Attributes for La Honda Geologic Map Units

Unit	Material Type (ref. 6: fig. 1, Table 1)	Pixels n	Pixels %	ALT. m	SLOPE deg.	ER %	SDC no/km^2	CURV deg.
S2	Colluvium	3589	2	101	12.9	0.33	162	19.0
S1	Coarse fan, terrace	3707	2	77	12.0	0.31	162	17.0
C1	Mudstone, siltstone	7729	4	154	14.4	0.45	165	19.1
F	Sandstone (ss)	5398	3	138	16.0	0.42	168	19.2
J1	Shale & mudstone	29,615	17	176	17.2	0.45	191	23.5
E	Clayey sandstone	57,772	33	197	15.9	0.44	167	20.8
J2	Shale & mudstone	12,896	7	140	18.4	0.37	180	22.1
C2	Mudstone, siltstone	14,641	8	309	16.4	0.43	167	19.5
K	Basaltic rock	12,310	7	282	19.1	0.40	181	22.6
R	Ss, minor shale	10,712	6	304	24.7	0.40	174	32.4
M	Sandstone & shale	5870	3	464	20.7	0.31	158	24.6

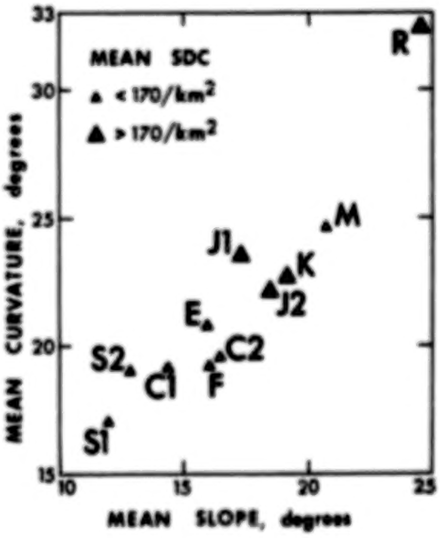


Figure 7. (left) Mean curvature (and frequency of slope reversal, mean SDC: triangles) as a linear function of mean slope angle for 11 La Honda map units (Table 1), from image processing the DEM (figs. 6, 4, & fig. 11 in ref. 7). Similar map units (S1, 2, etc.) tend to have similar terrain properties (ref. 6).

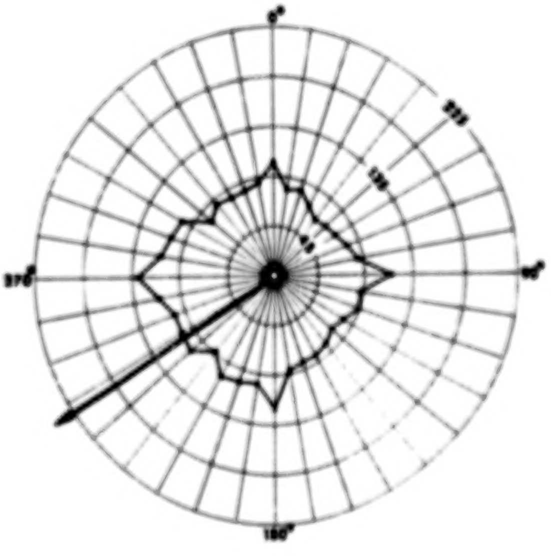


Figure 8. (right) Distribution of terrain aspect in the La Honda quadrangle, by 10-degree bins. Data are pixel counts weighted by slope angle (units shown are x 1000). Arrow, the vector sum, is normal to strike of coast and the San Andreas fault.

topographic characteristics. Few of the means in figure 7, however, differ significantly (dispersion values are large). Thus while topographic form may covary with attributes of the underlying materials in this area, the relation is weak.

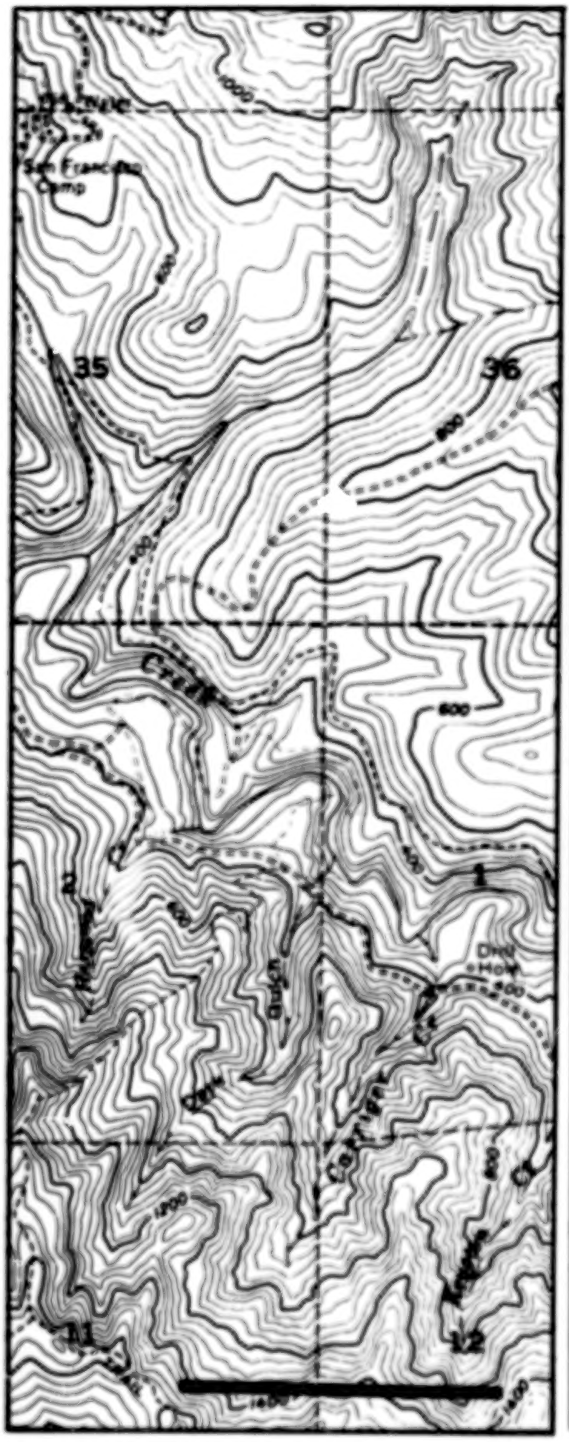
Aspect-frequency plots of slope-azimuth data are another useful product of image-processed DEMs (aspect map not shown) (4). Our rose diagram of pixel count weighted by slope angle (fig. 8), like that derived for the La Honda quadrangle from a TIN (fig. 8 in 7), improves upon previous aspect plots (4) by removing the statistical noise introduced by low slopes. The vector sum in figure 8, 238°, is nearly identical in both strength and direction to that from TIN analysis of the same data. The maxima in the cardinal directions result from as-yet unresolved artifacts of the x-y data structure.

4. THE TRIANGULATED IRREGULAR NETWORK

The TIN is a thrifty and variable-format structure for representing topography (3, 5). TINs now can be derived from DEMs automatically instead of manually (3, 13), but their full potential has yet to be realized for GIS analysis. We have modeled La Honda terrain (fig. 9) as a mesh of contiguous, near-planar TIN facets that are surface specific; i.e. they vary in size, (triangular) shape, altitude, and azimuth in response to local topography (3). Because these facets are surface specific units of land form, rather than arbitrarily sized and shaped windows or sample spaces, they are suited both for describing terrain and for making geomorphic interpretations.

The results in figures 10-12 come from ARC/TIN, software within the ARC/INFO system for GIS analysis (13), installed on a Prime 9955-II computer. This method identifies critical (information-rich) altitudes (5) at valleys, ridges, peaks, depressions, passes, and breaks-in-slope in a DEM by one of two search strategies. The VIP option (13) is a global and iterative (windowed) algorithm (fig. 11; ref. 7); the HIGHLOW option (13) makes a one-pass local search of the data (fig. 10). Next, the TIN procedure links resulting points in a triangular mesh (fig. 12), builds a topology describing their connectivity, and finally constructs tables of attributes that describe triangle geometry (see Table 1 in 7).

The HIGHLOW and VIP algorithms, which are designed for high-relief terrain and flatter areas, respectively, generate different TINs (figs. 11, 12) from the same source DEM (fig. 9). The former option appears to yield more small and slender triangles (fig. 12; n= 44,254) than the latter (fig. 11), in both cases perhaps a denser mesh than seems to us appropriate for the La Honda terrain at most scales. Tests need to be devised for quantitatively evaluating these TINs against the original topography (fig. 9). Similarly, TIN drainageway and ridge networks should be compared with those derived by image processing and other methods (cf. figs. 3 & 10; ref. 5).


San Francisco Camp
35
36
800
600
Creek
2
1
1200
11
12

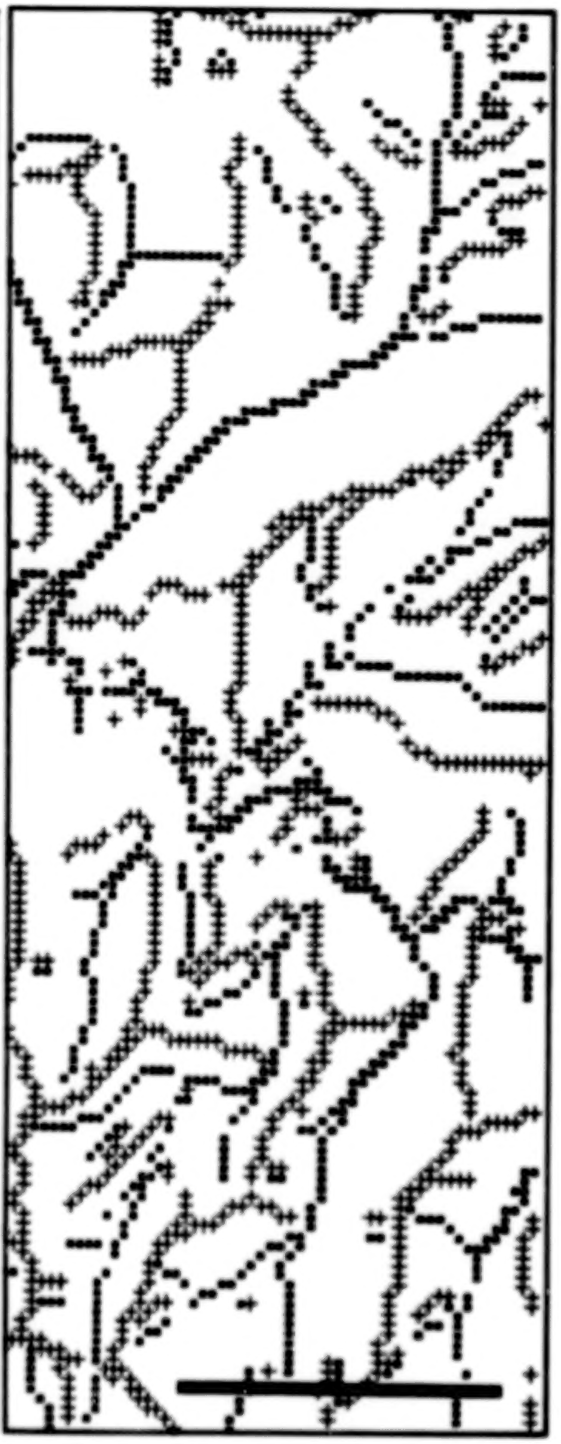
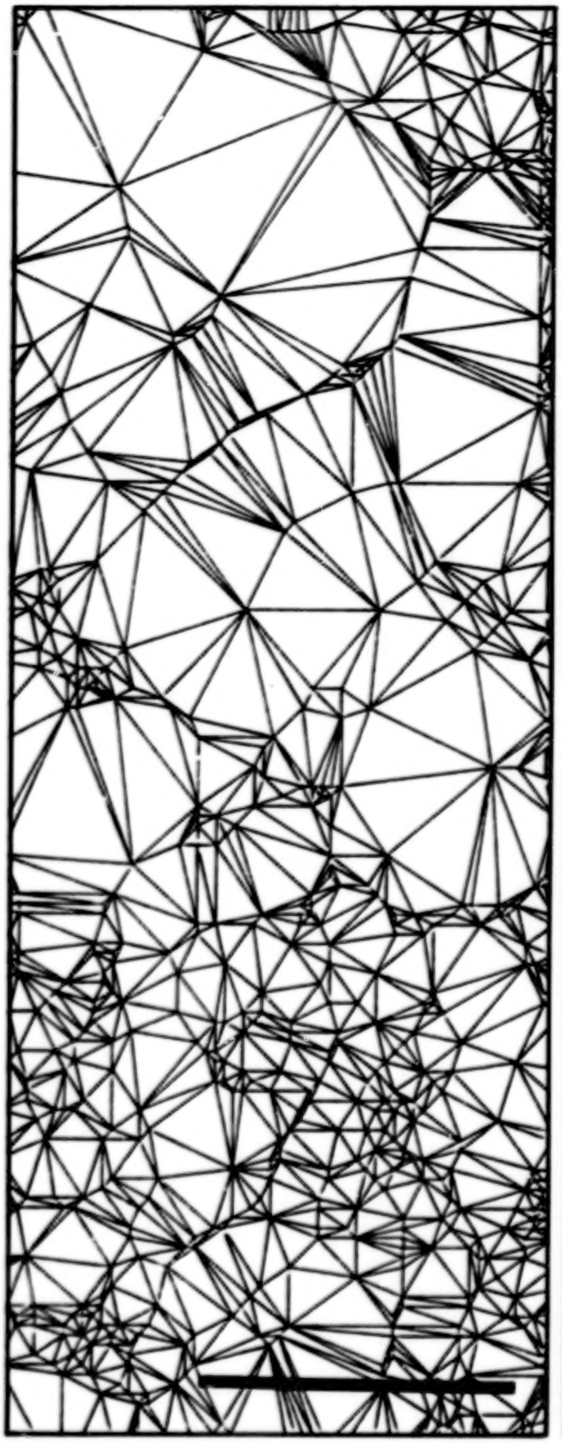


Figure 9. (left) Topography in SE corner of the La Honda quadrangle. Contour interval is 40 feet. Area in figures 9-12 is the same. Length of scale bar is 1 km.

Figure 10. (right) Major ridges (crosses) and valleys (dots) in the SE corner of the La Honda area, extracted from the DEM by TIN-based HIGHLOW algorithm (13). Compare with figure 3.



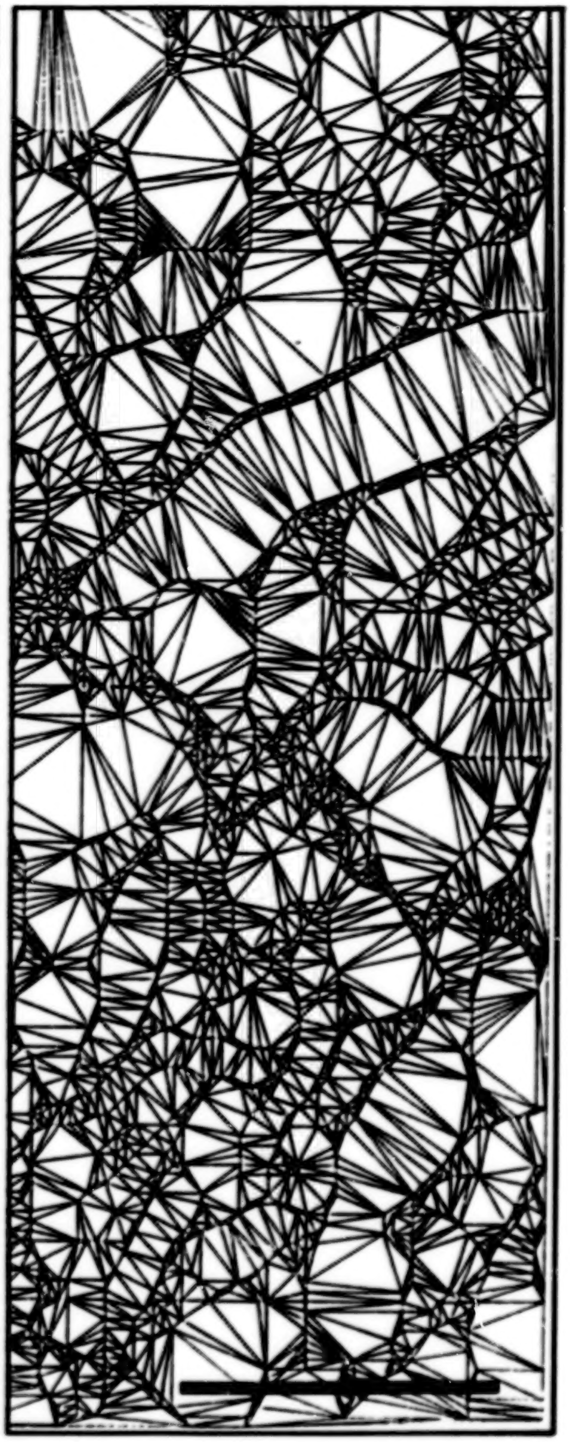


Figure 11. (left) TIN, without edge effects, in SE part of the La Honda quadrangle, from a 6% sampling of DEM altitudes processed through the VIP algorithm (13). Scale bar is 1 km long.

Figure 12. (right) Raw TIN for SE part of the La Honda area, resulting from HIGHLOW algorithm (13). Map contains edge effects. Triangle density is twice that in Figure 11.

Figure 11 shows part of a revised TIN (by the VIP option) from which the edge effects evident in (7) were removed by first adding the adjacent DEMs. The only important change is a 9% increase in the number of triangles. Comparison of the new TIN with our first map (7, fig. 4) shows contrasts only in triangles near the edges. The revised 6% sampling of the 172,980-altitude La Honda matrix resulted in 11,367 [old results: 10,438] critical points and 22,388 [20,853] triangles whose geometric mean area is 3711 [3674] m^2 and whose mean (unweighted) slope angle is 20° [20°]. Derivative maps and overall statistics and graphic analyses for the two TINs (e.g., ref. 7, Tables 2 & 3) do not differ significantly because edge triangles are a small percentage of the entire TIN.

The descriptive data computed automatically for each TIN facet can be handled like tabular information derived from image processing. Figure 13, a log area/slope angle plot from the VIP results in fig. 11, is a crude topographic signature for the La Honda quadrangle. It improves upon that in (7) by normalizing the asymmetric area data. No attempt was made to transform slope values, which are less badly skewed. Individual area/slope plots for the map units in Table 1 might yield different relations between triangle slope and size, which could reflect different material properties.

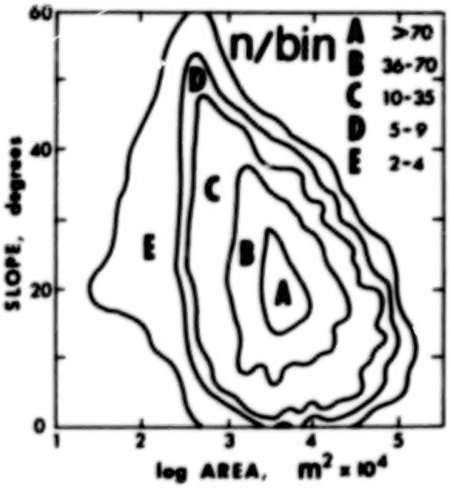


Figure 13. Slope angle as a function of log triangle area for 22,388 TIN facets in the La Honda quadrangle. Contours show frequency of occupancy of 2166 slope:area bins 1.6° high x log 1800 m^2 wide.

5. SUMMARY AND CONCLUSIONS

Advances in DEM quality and analytic techniques are combined in experiments with 1:24,000-scale topography in Central California. This paper samples some of the maps obtained, using automated TINs and image processing, from a nominally error-free 30-m 7.5' DEM and its derivatives. The new maps suggest that a topographic base for GIS analysis can be more comprehensive than what is available now. This terrain foundation, of which a raster of digital altitudes and a contour map are only the two most primitive ingredients, may include any of the variables used here and in (7). GIS coverages for terrain can be created either on TIN-based units or pixel-by-pixel, and can be treated as continuous data or as discrete classes.

This pilot study just begins to probe the opportunities and problems posed by the numerical analysis of topography within a GIS. Even such complex parameters as topographic grain (1) may be automated by image-processing of DEM data. Moreover, slope and aspect can be combined (e.g., 10) into terrain facets that mimic or surpass the TIN units. Further, what are the optimal levels of complexity for various TINs, and can TIN fidelity to the terrain be judged by other than visual inspection? Several coverages can be synthesized by multivariate methods to yield topographic signatures (8) and physiographic subdivisions (1). Current work is addressing these and other issues, especially comparing results from TIN and image processing and assessing their merits and drawbacks.

Finally, we advocate using computer maps of altitude derivatives to study slope processes, notably landsliding, and other aspects of the topographic evolution of San Mateo County (2, 6) and elsewhere (4, 5). The ability to manipulate altitude matrices quickly by computer allows emphasis to be placed on landscape interpretation rather than mere quantification. Our results to date suggest that creating a GIS and analyzing DEMs by image processing and TIN techniques will contribute significantly to solving geomorphic problems.

Acknowledgments

Stephen Ellen and John Tinsley clarified some of the ideas presented here. We thank Earl Brabb, Leonard Gaydos, and Richard Swinnerton for their continuing support and encouragement.

References

1. PIKE, R.J., 1972: Q-Mode Landform Regions of Southern New England. International Geography 1972, Proc. 22nd Int'l. Geogr. Congr. Montreal, Univ. Toronto Press: 365-367;

2. BRABB, E.E., 1987: Analyzing and Portraying Geologic and Cartographic Information for Land-Use Planning, Emergency Response, and Decisionmaking in San Mateo County, California. Proc. 2nd Annual Conf. GIS'87, San Francisco, ASPRS/ACSM: 362-374;

3. PEUCKER, T.K., FOWLER, R.J., LITTLE, J.J., and MARK, D.M., 1978: The Triangulated Irregular Network. Proc. DTM Symposium, St. Louis, Missouri, ASP/ACSM: 516-540;

4. EVANS, I.S., 1980: An Integrated System of Terrain Analysis and Slope Mapping. Z. Geomorph. N.F., Suppl.-Bd. 36, 274-296;

5. DOUGLAS, D.H., 1986: Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model. Cartographica, 23: 29-61;

6. WENTWORTH, C.M., ELLEN, S.D., & MARK, R.K., 1987: Improved Analysis of Regional Engineering Geology Using Geographic Information Systems. Proc. 2nd Annual Conf., GIS'87 - San Francisco, ASPRS/ACSM, 636-649;

7. PIKE, R.J., THELIN, G.P., and ACEVEDO, W., 1987: A Topographic Base for GIS from Automated TINs and Image-processed DEMs. Proc. 2nd Annual Conf., GIS '87 - San Francisco, ASPRS/ACSM: 340-351;

8. PIKE, R.J., 1987: Toward Geometric Signatures for Planetary Terrain: An Assessment of Earth at 1:24,000 Scale. Lunar & Planet. Science XVIII. Lunar & Planetary Inst., Houston: 780-781;

9. BATSON, R.M., EDWARDS, K., & ELIASON, E.M., 1975: Computer-Generated Shaded-Relief Images. Jour. Research USGS, 3: 401-408;

10. LAMB, A.D., MALAN, O.G., & MERRY, C.L., 1987: Application of Image Processing Techniques to Digital Elevation Models of Southern Africa. So. African Journal of Science, 83: 43-47 and cover;

11. ESL (Electromagnetic Systems Laboratory, Inc.), 1983: IDIMS Users' Guide, Volume 1. Sunnyvale, California;

12. PIKE, R.J., and WILSON, S.E., 1971: Elevation-relief Ratio, Hypsometric Integral, and Geomorphic Area-Altitude Analysis. Geol. Soc. America Bull., 81: 1079-1084;

13. ESRI (Environmental Systems Research Institute, Inc.), 1987: TIN Users' Manual. Redlands, California.

UPDATING MULTI-LAYER VECTOR DATABASES

Jonathan Corson-Rikert
Laboratory for Computer Graphics and Spatial Analysis
Graduate School of Design, Harvard University
Cambridge, Massachusetts 02138

ABSTRACT

Recent advances in map digitizing software and in map scanning hardware are beginning to reduce the time and costs involved in building geographic data bases. Maintaining the spatial components of a geographic data base, however, can involve quite diferent and equally demanding system requirements. Editing in one data layer may affect the currency of other layers, and procedures must be developed so that change can be propogated consistently across an entire data base. This paper reviews briefly the range of institutional and data-dependent considerations affecting updating, and discusses software developments underway at the Laboratory to support consistent geographic data base maintenance.

1. INTRODUCTION

The Laboratory for Computer Graphics is currently developing new geographic information system software as part of the GRASSLANDS project supported by the American Farmland Trust. This software will include new input and editing software called ROOTS, initially developed as the TRACE program with support from the International Business Machines Corporation (Corson-Rikert and White, 1985b). The GRASSLANDS software will include with ROOTS a UNIX™ version of the vector processing and display capabilities of the Laboratory's ODYSSEY geographic information system currently installed at some 40 universities around the world. The software will be compatible with and extend the vector input and processing capabilities of the GRASS software developed at the U.S. Army Construction Engineering Research Laboratory in Champaign, Illinois. The GRASS system already supports advanced image processing and raster-based analysis capabilities and is installed a number of locations (Westervelt, Goran, and Shapiro, 1985).

While the initial motivation for developing ROOTS came from a need for a fully interactive program for initial digital map overlay entry, we have recognized the need to support editing and updating as well. Once a data base has been built, display and analysis software become more important for producing final map products. But adding new data and maintaining existing layers must continue, and software tools need to support the somewhat different requirements of data maintenance. This problem is magnified if a data base is initially built under outside contract and transferred to a unit of government or other user group for ongoing maintenance. The quality of the data may suffer rapidly if improper maintenance corrupts good data, and staff will find maintenance prohibitively complex and frustrating.

Inertia will always tend to work against updating. The time required to collect and check current data may approach the costs of original data compilation, especially when several operational units of government are involved. Maintenance ideally includes keeping tracking successive changes and monitoring the ongoing quality of spatial information as well as storing the latest information itself.

This paper will discuss briefly a range of issues in database updating, from institutional considerations and data-dependent effects to alternative strategies for software tools. While the updating of attribute records associated with database layers will be discussed, the primary focus of the paper is on management of the spatial component of data.

2. DATABASE DESIGN AND MANAGEMENT ISSUES

Multiple-layer databases characteristically include layers of great disparity in scale, accuracy, and consistency. Planning the successful coordination of diverse layers requires both a good spatial framework and careful organization of data layers. The spatial framework typically includes a network of precisely defined and coordinated control points identifiable both on the ground and on as many source layers as possible. A base map must then be assembled at a consistent level of accuracy for the study area, including features such as roads and water bodies which serve as references for thematic layers. Ideally the base map series would be derived by photogrammetric compilation from large scale aerial photography; for regional studies the U.S.G.S. quadrangle sheets or their digital representation may prove adequately reliable. When a digital base map has been prepared, thematic overlay layers may be digitized or scanned into the same coordinate system by use of the control points common to all layers.

Many layers share some features with other thematic coverages . A feature such as a river or lake shoreline will appear in a water layer but will almost certainly also occur in soils, parcels, zoning, vegetation, and land use layers. For reasons of consistency and clarity, the feature should be identical each time it occurs in the final database even though there will be as many original versions as source layers upon which it appears. The definitive version of the feature should be identified in the planning phase, entered once, and copied intact to all the layers. When that feature requires updating, the changes should be made first on the definitive source and transferred across all other affected layers (figure 1).

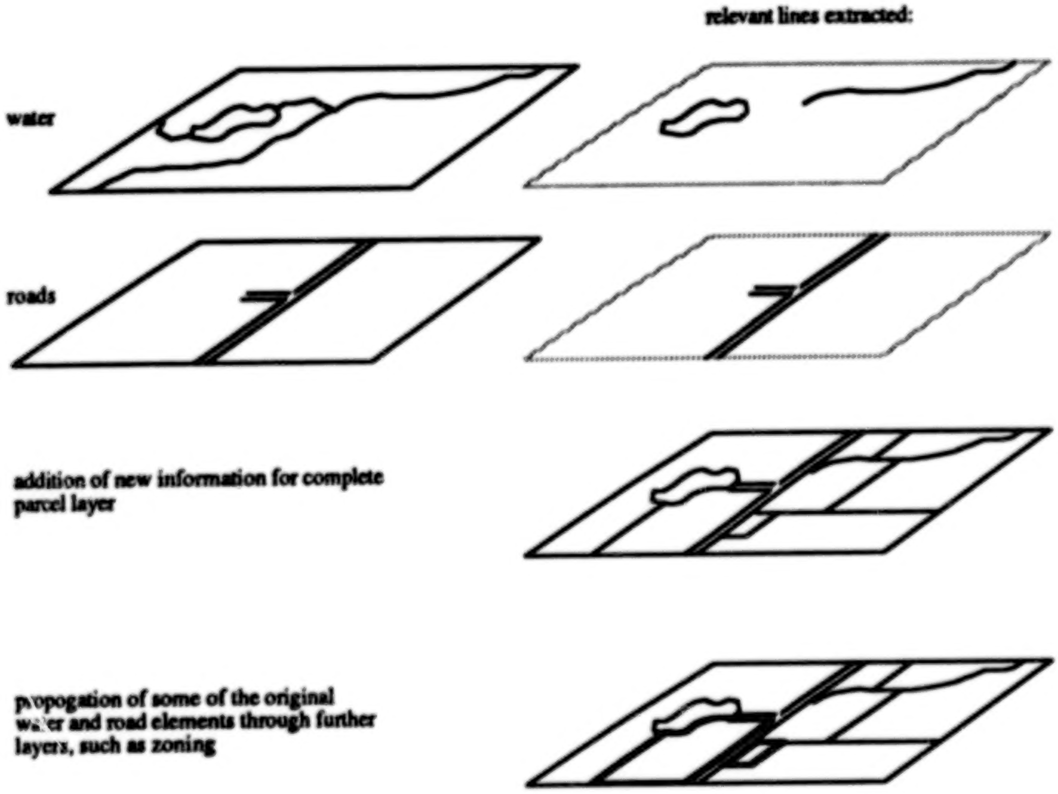

relevant lines extracted:
water
roads
addition of new information for complete parcel layer
propogation of some of the original water and road elements through further layers, such as zoning

Figure 1. Extracting elements from primary sources to serve as boundaries in other layers

When the relationships between layers are more extensive, the management of both construction and updating may change. Zoning boundaries are often defined to follow parcel boundaries wherever possible; a zoning layer may therefore be nearly completely dependent on the parcel layer. In such cases the ability to interactively copy spatial data between layers may prove more efficient than file-based transfers.

Other general conditions may affect database construction and editing strategies as well. Ownership of or responsibility for data may vary from local to federal levels. When an agency has responsibility for data maintenance and archiving it can be less flexible in managing that data in concert with other layers.

Some data layers are inherently independent, with few elements in common with other layers. Elevation data are typically generated by specialized photogrammetric equipment and displayed on source maps as contour symbolism that has only minimal graphic interaction with other layers. Contour lines do not normally serve as boundaries in other coverages, so the elevation layer can typically be digitized and updated independently of other layers.

The inherent scale and accuracy constraints of a data layer will also affect how that layer will be incorporated into a database and how it will be managed once entered.

For example, soil information is typically one of the less accurate and more generalized layers in a database. Samples used to compile the information are limited in number and are interpolated to soil polygons using subjective expertise; classifications are often generalized; and the data are then compiled on non-planimetric photo base maps which provide useful reference information but do not enhance the spatial integrity of the data. Soil information often cannot be correlated exactly with other layers except at physical features such as water bodies, where the soil layer must often be edited to match more reliable base information.

Because soil information is critical for a number of analyses, however, and because it can also play a role in defining such tangible issues as assessed value or eligibility for federal agricultural programs, a special effort is often made to relate soils information to other layers using both manual recompilation and scanning and editing techniques. (Nystrom et. al.,1986; Chrisman, 1986).

In some cases the official designation of a resource feature may be more important in a database than the best available naturally defined boundary. Once a wetland has been defined in any official way, the official version rather than the changing or more precisely identifiable natural boundary is used in permit granting or other land use decisions. The availability of better information may not be cause for updating.

Much of the credibility of a database depends on consistency and completeness rather than absolute accuracy. A highly visible flaw such as a missing new shopping center or residential development can undermine confidence in an otherwise complete and accurate presentation map. Failure to be able to explain how a composite overlay has been constructed or an analysis accomplished can also limit the effectiveness of a database in documenting resource conditions or recommending land-management options.

In summary, cross-relationships and priorities should be defined as explicitly as possible prior to construction of a database. A hierarchy of data elements and their sources needs to be designed, constructed, and maintained. If relationships among data layers are well understood and well documented, input will flow more smoothly, and ongoing change can be accomplished without a loss of quality.

3. OVERLAY TECHNIQUES FOR EDITING AND UPDATING

The simplest form of overlay is purely graphic. While this capability originated with manual systems pioneered by Ian McHarg and others, the ability to display many source layers at a common scale on a computer screen has dramatically increased the practical application of multiple-layer techniques. Although the relationships between layers exist only in the image, a graphic overlay tool can be adequate for a significant range of display functions. Not the least advantage is that purely graphic overlay systems often support text annotation layers as well. The composite image on the screen can be saved and enhanced by manually adding color shading, legends, and labels. Programs such as AutoCAD™ (Autodesk, Inc.) and MacPaint™ and MacDraw ™ (both Apple Computer, Inc.) can produce sophisticated output from simple data structures, but require the user to manage all the graphic enhancement and labelling manually. Making shading patterns and applying them manually to 20 or 30 polygons is a simple task, but doing the same for 1500 polygons is much less appealing.

The disadvantages of graphic overlay lie more with what it does not do. If editing is required, the same edit must be performed in each layer consistently. Since no analysis of the interaction between the layers is performed, no information is gained beyond what is graphically apparent. For example, the intersection of a floodplain and a parcel can be seen but not quantified. For determining the area of intersection or reporting the total intersection of the floodplain and certain parcel types across a while map, more powerful overlay capabilites are required.

The ODYSSEY geographic information system and a number of other software systems using topological data structures offer the capability to perform polygon overlay processing. The overlay processing is performed using the topologically coded polygon boundary segments (called chains or arcs) between the points where polygons intersect (called nodes) (figure 2). The chains are coded with left and right polygon identifiers which are maintained through the polygon overlay processing so that the original source polygons in each coverage can be traced. The polygon overlay capability has many applications for updating as well for generating a new layer from the intersections of two existing layers. The most obvious updating application is to compare the same coverage for two different time periods, to identify areas of change (Chrisman, 1982). Alternatively, known changes can be overlaid with an original file prior to an automated or interactive editing process to merge in new data and purge the old.

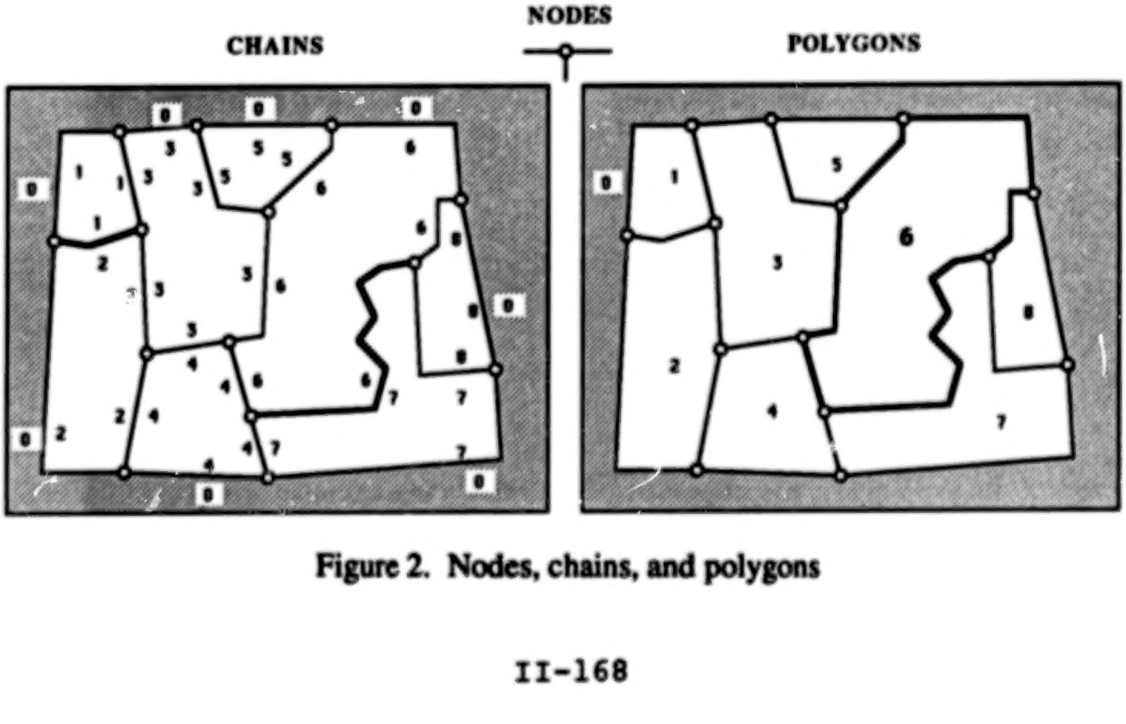


Figure 2. Nodes, chains, and polygons

Polygon overlay works by constructing the geometric intersections of two layers and producing a third layer that is the spatial composite of the two parent layers. The process is performed on two chain files and produces a chain file result that has new polygon left and right identifiers which can be linked via a cross-reference file to each parent coverage. Editing takes place on the output file using the information in the cross-reference attribute file.

In many cases the desired editing can be performed by predetermined and well-defined procedures which use both the spatial and attribute information gained from polygon overlay. Procedural analyses or editing functions are applied in batch mode to an entire file without intervention from the user. The spatial and attribute characteristics of the input files and of the overlay can be quite complex and must be well understood to be meaningfully used. A simple example would be reporting the areas of intersection between a floodplain coverage and a simple map of developed vs. undeveloped areas in a town (figure 3). Four cases will exist in the intersection file: developed land in the floodplain, developed land out of the floodplain, undeveloped land in the floodplain, and undeveloped land out of the floodplain. The cross-reference file produced by the ODYSSEY intersection process presents the information in the following table:

output polygon id	floodplain	developed	perimeter	area
neither	no	no	ft.	sq. ft.
only floodplain	yes	no	ft.	sq. ft.
only developed	no	yes	ft.	sq. ft.
both	yes	yes	ft.	sq. ft.

In this simple a case, the cross-reference file accomplishes the analysis and reporting procedure directly. When the two original files have a more complex attribute structure, the cases in the output cross-reference file become more complex, and the desired output information must be selected or aggregated from the cross-reference file.

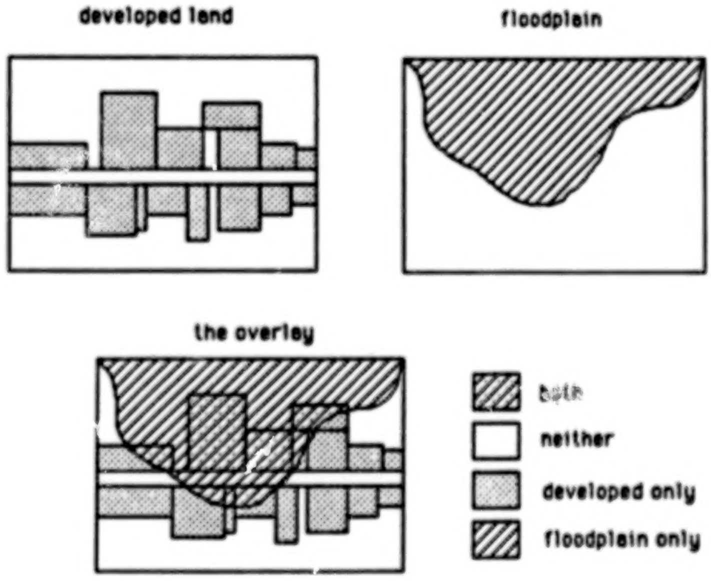


Figure 3. A simple polygon overlay example

A more complex example used at the Laboratory was updating a land-use/land-cover file for the Minute Man National Historical Park in Concord and Lexington, Massachusetts (Corson-Rikert & White, 1986a). The original compilation in 1984 had outdated road information that needed to be updated to reflect road closings in the Park and new subdivision roads in developing areas of the towns, called 1985 roads. Polygon overlay was performed using the two input chain files, 1984 land-use and 1985 roads, which both represented roads as areas between parallel lines (double-line streets) and not as simple linear features. The resultant spatial intersection chain file was then modified using information in the cross-reference file to produce a final 1985 land use file.

A number of specific changes needed to occur in the intersection file based on attribute conditions in the two input files. Areas that were not roads in the 1984 coverage but which were in 1985 needed to be coded as roads in the output file. Areas which were roads in 1984 but not in 1985 needed to be coded as ex-roads in the output file (more on this code in a minute). Areas that were roads in both or in neither input coverage needed to be left alone. At the Laboratory, these coding rules are then built into command files for the Harmony program (White, 1985), which will step through a chain file and recode the left or right I.D. if one of the above change rules applies. This process results in a chain file with correctly coded left and right polygon I.D.s except for the special cases of the ex-roads. Where an ex-road borders a single non-road polygon (and if the user knows that) the recoding is straightforward, but when several different land use polygons abut the ex-road, the recoding must be accomplished on a case-by-case basis by interactive editing. The recoded chain file then needs an additional processing step to remove chains with identical left-right identifiers (both road), which can be accomplished by Harmony at the time of recoding or by a separate pass in ODYSSEY.

Polygon overlay can accomplish very large and complex updating tasks and is well suited to production environments where the coding changes can be planned in advance and stored in command files. The problems alluded to in managing the recoding of the ex-road attributes indicate limits to the process, however. When such problem cases are combined with the effects of different tolerance parameters for reducing sliver gaps and overlaps, even an experienced user can quickly create a mess that is hard to unravel. These problems become more unmanagable when placed in the hands of users not familiar with chain topology or with overlay concepts. Batch procedures are best defined by system managers who control the operations performed and the circumstances of use very tightly.

4. INTERACTIVE TECHNIQUES FOR UPDATING

Our own experience at the Laboratory has lead us to support more and more forms of interactive editing and updating with increasing attention focused on the case of managing information across more than one layer. We find that this more interactive approach to editing has great utility as a supplement to, and at times an alternative to, the more procedurally defined processes of selection, aggregation, or polygon overlay.

The principles of graphic interaction supported in the ROOTS program have been discussed in a previous paper (Corson-Rikert and White 1986b), and need only be briefly mentioned here. File creation and editing is done with constant feedback on the screen, and the topological structure is updated as each point is added or moved or deleted. The user can concentrate on correctly capturing the graphic source with the current state of the data base immediately evident. Graphic windowing, attribute assignment, interactive polygon naming, and other features are combined to provide a flexible and powerful input and editing environment. The program reads and writes data in ODYSSEY chain format, or will optionally produce polygon output or convert from vector to raster format for input to the Map Analysis Package (Tomlin, 1983) and GRASS.

One common application can illustrate the different advantages of a procedural editing process and more interactive graphic editing. The ZIPPER program for edge matching two ODYSSEY chain files, developed at the University of Wisconsin-Madison (Beard and Chrisman, 1986), will significantly reduce the manual editing needed to edge match two adjacent map sheets using a tolerance distance that decreases in proportion to distance from the common edge. This process results in smoother transitions across edges and limits manual editing to lines greater than the tolerance distance apart along the edge. Whether additional editing is required or not to complete the edge match depends on the consistency of the two files and on the tolerance distance used.

An alternative approach with the ROOTS program is to append the second file to the first. Any points along the edge within the current tolerance setting are merged as the second map is read in; smoothing the visual transitions between sheets and merging any lines still disconnected is then accomplished interactively. If a chain is left dangling, it can be merged graphically by dragging the end node to the desired position at another node or along another chain. Individual points along a chain can be moved by pointing and dragging the adjacent line segments. The edge can be matched to the user's satisfaction from within ROOTS as a single process.

A combination of the two approaches could in this case take advantage of the strengths of both. By using ZIPPER to perform the initial edge match and reading the results into ROOTS, remaining gaps could be interactively fixed very quickly. On large files the additional step of running the ZIPPER process would be offset by the savings in editing time.

We have already incorporated an option for graphic overlay in the ROOTS program by supporting up to four reference files in addition to the currently active file. The reference files are drawn in dashed mode or in a different color each time a new window is set, and serves as a reference for placing new information or correcting the alignment of the current layer with the pre-existing reference file. This feature is also useful for showing a polygon layer as a reference when digitizing point symbols.

One simple set of extensions beyond graphic overlay would be to facilitate alignment of the current layer with the reference file. Currently each new point is checked against all other points and line segments in the main file to see if a point or line is matched or crossed. Those tests could be extended to search the reference file as well, so that a complex boundary could be traced with assurance that the new version will align correctly, or a parcel boundary could be brought up against a road in the road file and the boundary ended precisely at the road. The tests could even be extended to disallow certain cases, such as a contour line crossing a river or a house crossing a parcel line.

It might well be easier for the user to support a pick and copy option, where a road which serves as a parcel boundary could be copied from the road reference file to the active parcel file by picking it. The danger of this approach is that new information will duplicate existing information in the active file. Additional screening can be implemented to discard duplicate information, but the overhead involved may degrade the performance of the software significantly.

When building a new database, the hierarchichal method of data organization has facilitated transfer of a single version of a common element such as a road or stream from the base layer to start subsequent data layers such as parcels or land use. This approach has an added advantage in the ROOTS priority management system, where data read in first have the highest priority. New data are always matched to existing so that the first version of a node or point is maintained fixed unless delibrerately moved. Similarly, when a new line terminates at or crosses a line segment, the intersection is placed exactly on that segment. A long straight road or a map sheet boundary intersected by many other chains as the new layer is built would remain straight in the final file even though broken into several segments.

The updating case is more complex. When several layers are interdependent, as described in the initial sections of this paper, changes must not only be applied to one file but also be consistently propogated across any other file incorporating that same spatial element. A new highway bypass will affect many layers including roads, soils (cut and fill material), land use/land cover, parcels, and zoning. Because the new road outline will interact with different layers differently, the new chains cannot be copied intact from one file to another. The problem may be compounded in the case of a series of interconnected files; the road will affect the parcel layer, and the composite changes in both those files will affect the zoning layer.

One management solution would be the composite file or least-common geographic unit (LCGU) file. Essentially the file would resemble an interactively built polygon overlay intersection file. All layers which interact would be entered into one file (or perhaps linked to one parent data structure). New lines would be cut into pieces as they intersect other lines, but additional overhead of attribute infomation would be required to indicate in which layers the line is active. A section of road could be broken into ten pieces by parcel intersections but could be treated as a unit if roads were the currently active attribute. Intersections with parcel lines would be transparent to the road layer, in much the same way that chain segments are treated as a inseparable units of a normal chain. One can even imagine maintaining separate polygon lists for each attribute layer, although the tests to confirm all possible effects of even a minor editing change could become ponderous.

The composite approach could be simplified by a method similar to processing the results of a polygon overlay, as described above for the 1984 and 1985 roads example. The individual layers could be extracted from the composite file by line attribute selection, and then processed independently as more managable files. No changes could be permitted to the spatial data after separation, however, or the composite file would become outdated. A polygon overlay process could rebuild the intersecting chains, but rebuilding the attribute structure for all the layers would be difficult.

A number of problems argue against a single composite approach, the most obvious being the overhead requirements. Another major issue is that all layers which interact do not necessarily interact at the same tolerance levels. One could easily imagine a situation of a loosely defined soils boundary being aligned to a more precise boundary from the parcel map because of accidental proximity. Layers would be forced into alignment when that alignment would be meaningless or misleading on the ground, since in our example the soil polygon would rarely if ever coincide with the parcel file. Adding explicit priority management for each layer would only increase enormous overhead requirements and not even solve the problem completely. Ultimately, the problem with the composite approach is that it forces all parts of many layers to interact when only certain common parts of the layers(water boundaries, for example) have any meaningful relationship on the ground. This distinction again separates a geographic information system from standard multi-layer CAD graphic overlay programs used in automated mapping/facilities management applications.

A more discrete management capability would be preferable for administrative, data integrity and security reasons. The composite file would face the same issue that must be addressed with polygon overlay -- If you increase your error tolerance to eliminate undesirable slivers, whatever precision may have existed in some of the layers will be degraded. The tiny intersection units in a composite file would likely have very little correspondence to the situation on the ground.

From a user's perspective, the most important capability is to be able to bring known changes to each layer of the database in succession in an orderly fashion. We have found that students as users relate very well to the hierarchical concept of database organization, in which base information is copied to derivative layers to maintain consistency throughout the database. What users need for updating is a clear process for incorporating changes in these structured layers.

The prototype for change management now exists in ROOTS. When a file is brought in to the program for editing, any changes to spatial or attribute data are labeled as changes, and at the user's option can be written out separately from the main file as well as stored with the latest copy of the main file.

Under this system, the highest file in the hierarchy of the database that required editing should be edited first -- roads in the base map, for example. When changes are complete, the affected chains and new data are written out as a separate changes file. That file can be read as a reference file into a derivative coverage such as parcels. With appropriate color reinforcement to distinguish old and new, obsolete data can be removed and the new data appended into the file from the changes file, and the line or polygon attributes for the affected area interactively reviewed. The process can be repeated if changes in the parcel file needed to be passed on to derivative coverages of parcels such as zoning.

We feel this approach has one major advantage - simplicity. The updating process would closely resemble traditional drafting procedures as a model for change, which improves the likelihood that users without extensive system management experience could understand and internalize the process and manage updating on their own. Changes would be reflected graphically, and the topological maintenance capabilities of the software would provide assurance at each step that the changes were being successfully integrated.

In conclusion, we emphasize our goal of providing software tools that will encourage the people closest to the source data to manage that data themselves. Efficient and satisfying tools which allow for flexible approaches and provide good feedback can speed up the process and improve the quality of data base. If the person entering the data knows the data, issues of interpretation and consistency between layers will be resolved correctly at the time of original entry or editing. In all likelihood the time for a knowledgable person to digitize or edit will be no greater than that person's time required for a thorough check of another's work, with the added advantage that the person who will use and make decisions with the database will have first hand knowledge of its strengths and limitations.

References

1. Beard, M.K. and Chrisman, N.R. 1986, "Zipping: New Software for Merging Map Sheets", Proceedings, American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing 1986 Annual Convention, Volume 1, pp. 153-161.

2. Chrisman, N.R. 1982, "Methods of Spatial Analysis Based on Error in Categorical Maps", unpublished Ph.D. thesis, University of Bristol.

3. Chrisman, N.R. 1986, "Effective Digitizing: Advances in Software and Hardware", Proceedings, American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing 1986 Annual Convention, Volume 1, pp. 162-171.

4. Corson-Rikert, J. and White, D. 1985a, "Issues in Map Digitizing", Laboratory for Computer Graphics and Spatial Analysis, Report Number LCGSA-85-17.

5. Corson-Rikert, J. and White, D. 1985b, "The TRACE Program for Map Digitizing",

Laboratory for Computer Graphics and Spatial Analysis, Report Number LCGSA-85-18.

6. Corson-Rikert, J. and White, D. 1986a, "Building Large Data Bases for Planning: The Minute Man National Historical Park", Proceedings, American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing 1986 Annual Convention, Volume 3, pp. 142-151.

7. Corson-Rikert, J. and White, D. 1986b, "Comprehensive Map Input Software: Building on the Minute Man National Historical Park Data Base Project", Proceedings, Urban & Regional Information Systems Association Annual Convention, Volume 1, pp. 190-202.

8. Department of Landscape Architecture, Graduate School of Design, Harvard University 1985, Alternative Futures for Minute Man National Historical Park., Cambridge, Massachusetts.

9. McLaren, R. and Brunner, W. 1986, "The Next Generation of Manual Data Capture and Editing Techniques - The Wild System 9 Approach", Proceedings, American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing 1986 Annual Convention, Volume 4, pp. 50-59.

10. Nystrom, D, Wright, B, Prisloe, M, Batten, L. 1986, "USGS/Connecticut Geographic Information System Project", Proceedings, American Congress on Surveying and Mapping - American Society for Photogrammetry and Remote Sensing 1986 Annual Convention, Volume 3, pp. 210-219.

11. Tomlin, C. Dana. 1983, "Digital Cartographic Modeling Techniques in Environmental Planning", Ph.D. dissertation, Yale University.

12. Tomlin, C. Dana. 1985, "The IBM Personal Computer Version of the Map Analysis Package", Laboratory for Computer Graphics and Spatial Analysis, Report Number LCGSA-85-16.

13. Westervelt, J., Goran, W., and Shapiro, M. 1985, "Design and Development of GRASS: The Geographical Resources Analysis Support System", U.S. Army Construction Engineering Research Laboratory, P.O. Box 4005, Champaign, IL 61820.

14. White, D. 1984, "Relief Shaded Thematic Mapping By Computer": The American Cartographer, Vol. 12, No. 1, pp. 62-67.

15. White, D. 1985, "A Generalized Cartographic Data Processing Program", Proceedings, Auto-Carto 7, pp. 560-569.

DATA BASE MANAGEMENT THEORY

BLANK PAGE

DATA STRUCTURES FOR QUADTREE-ADDRESSED ENTITIES ON A SPATIAL RELATIONAL DATABASE

Gregory A. Elmes

Department of Geology and Geography

West Virginia University, Morgantown, West Virgina

ABSTRACT

The performance of a geographic information system depends on the structure of geographic and cartographic data in a database management system and the formulation of search mechanisms to retrieve spatial entities and their attributes. Relational databases offer powerful and flexible means to handle large collections of data and a full range of geographic or cartographic entities. A quadtree block entity is defined and implemented on the GEOVIEW system. SQL query language macros are implemented for retrieval of quadtree entities for output to graphics and spatial modeling applications.

1. INTRODUCTION

Geographical information presents special problems to the design and construction of databases. Of the several available candidate structures, the relational database technology has the power and versatility to handle geographical data while benefiting from rapid technological advances. One commercial system (ARC-INFO) has partially adopted relational techniques but has dealt with the dual nature of spatial data by storing positional data separately in a non-relational form (1). Several theoretical relational data models have emerged which do not resort to a separate database for the locational data (2, 3, 4). Of these GEOVIEW (4, 5) claims to draw on the full power of the relational model as embodied in the commercial ORACLE database system supported by a subset of SQL query language. This study implements quadtree entity types in the GEOVIEW data model.

GEOVIEW is an input and edit system intended to provide the basic database requirement for any geographic system (4, p. 1). As such it is designed to incorporate an unlimited range of geographic entities and their attributes. Addition of new entity types to the database does not require changes in the underlying structure. The design permits the simultaneous handling of vector-based and of raster-based spatial data. GEOVIEW thus provides a testbed for prototyping multiform spatial data types. It has the major operational advantage that all permanent data, in and about the database, are held as relational tables (5).

Quadtree blocks, derived from the decomposition of grid images into homogeneous quadrants, are treated as a geographical entity type. Although they have no physical analogue in traditional cartography, quadtrees offer a useful means to manage certain categories of geographical information. In the course of implementing the quadtree block entity two research questions are addressed:

1) How should intrinsically hierarchical entities (e.g. quadtree blocks) be incorporated in a relational data structure?

2) What benefits and disadvantages accrue from the inclusion of the quadtree entity type in the GEOVIEW design?

The significance of these research questions derives in part from the intense research activity focusing on quadtrees as data structures in their own right. Quadtree structures are the bases of comprehensive geographical information systems (6, 7, 8, 9). The quadtree block is a useful geographical entity, but it is intrinsically interesting to adapt the relational structure to manage hierarchical entities for comparison with the representation of vectors in a hierarchical data structure.

Input and editing features of GEOVIEW are being developed concurrently with the management of vector and raster data types Several features of the GEOVIEW operating system were used in the production of results. These include the SQL control macros, the search and retrieval modules, graphics driver, and image manipulation. While several methods of quadtree block decomposition are available, a recursive algorithm was developed to provide specific window coordinates. Only one representation of quadtree blocks is currently fully implemented on the GEOVIEW system, although other structures are possible. Three grid-based data sets were available for experimentation: The Sussex Land Use Inventory; planning data compiled for Scotland by the Transport and Recreation Research Unit (TRRU); and a portion of a Landsat TM image.

2. QUADTREE BLOCK REPRESENTATION IN GEOVIEW ENTITY TABLES

Entities having an explicitly hierarchical relationship present problems to the relational database. If normal form is preserved, the result is considerable redundancy and impaired performance in CPU time and space requirements. Waugh and Healey suggest a pragmatic solution, one that sacrifices integrity of the normal form in favor of performance (5).

Using a standard cartesian coordinate system, the first method of quadtree block addressing requires the definition of a window containing the block by specifying the low and high X and Y coordinate values. The low and high coordinate pairs are then entered in the entity base table with the corresponding entity identification number for the block. The entity identification number is the primary key for the entity base table and the foreign key to the attribute base table.

A second form of addressing is provided by the properties of a family of space-filling curves known as Peano curves (10). The location and size of the cell or block of cells are specified when the coordinates of the Peano curve are encoded in the form of Morton addressing, resulting from the bit interleaving of the binary representation of the Y and X coordinate values. In the GEOVIEW design the Morton low address and high address are stored together with the block level (Table 1).

3. QUADTREE BLOCK IMAGES IN GEOVIEW

3.1 Sussex Land Use Inventory Data (SLUI)

The first of the three sets consists of nominal data at 200 meter resolution. Several significant aspects are raised concerning the processing of the SLUI test data set. Because each coverage is comprised of mutually exclusive classes, there

ENTITY_ ID	TYPE	DATA*	WIN_ X LOW	WIN_ Y LOW	WIN_ X HIGH	WIN_ Y HIGH
1	5		0	0	2	2
2	5		0	2	2	4
3	5		1	0	3	2
4	5		2	2	3	3
.	.		.	.	.	.
.	.		.	.	.	.
.	.		.	.	.	.

*Column of "LONG" data type containing XY coordinate string defining entity.

ENTITY_ ID	TYPE	DATA*	LINK	WIN_ LOW+	WIN_ HIGH+	LEVEL
1	5	B0		0000	0000	0
2	5	B1		0001	0001	0
3	5	B2		0002	0002	0
4	5	B3		0003	0003	0
5	5	B4-B7		0010	0013	1
6	5	B48-B63		0300	0333	2

+ base 4 Morton key

1	5	B0, B1 ...B63		0000	0333	3
2	5	D0, D1 ...D63		0000	0333	3

Table 1. Alternative Base Entity Table Configurations for Quadtree Block Storage.

was no additional loss of information in the process of encoding into quadtree blocks. An exception was the preparation of the land use data which was originally recorded in a hierarchical format, e.g. by land use order, class, sub-class, group, and sub-group.

The ORACLE database does permit alternative means to structure entities hierarchically. A scheme is being investigated to reduce the problem of redundant storage created by hierarchical data ordering.

The homogeneous blocks of the geological units (Figure 1) tend to be small because few neighboring pixels share the same values.

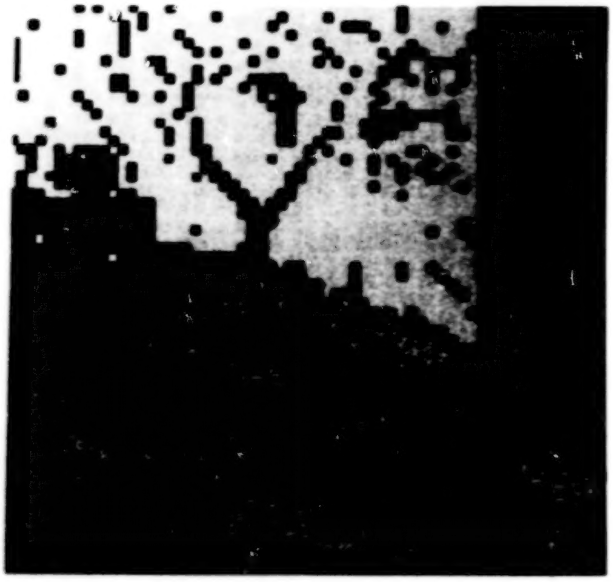


Figure 1. Geology Coverage, SLUI test data.

Note that the area in which the test data is embedded is not stored in the ORACLE database. In contrast, the areas of urban development (Figure 2) and the local government administrative districts are represented by comparatively small numbers of high order (level 2 and 3) blocks. The quadtree block method of data handling is an effective means to reduce storage space and processing times compared to the handling of the data as individual cells. Storage requirements were reduced by 1/3 to 1/2 over the original indexed sequential files.

3.2 LANDSAT TM Image of Northern Belize

The second example is of classified data with a 30 meter scale of resolution. For test purposes a 64 x 128 subset of the image was extracted. LANDSAT TM data is an example of form for which quadtrees are intrinsically unsuited. A case may be made for certain applications, e.g., a landuse classification stored for change detection purposes or georegistration with base maps. The image depicts many small regions, some of which are strongly linear in form (Figure 3). The linear features are stored as a large number of low order (level 0) blocks. Storing the data in classified form results in information loss and flexibility.

Different users may well have selected a different classification of the raw LANDSAT imagery. Yet, the GEOVIEW

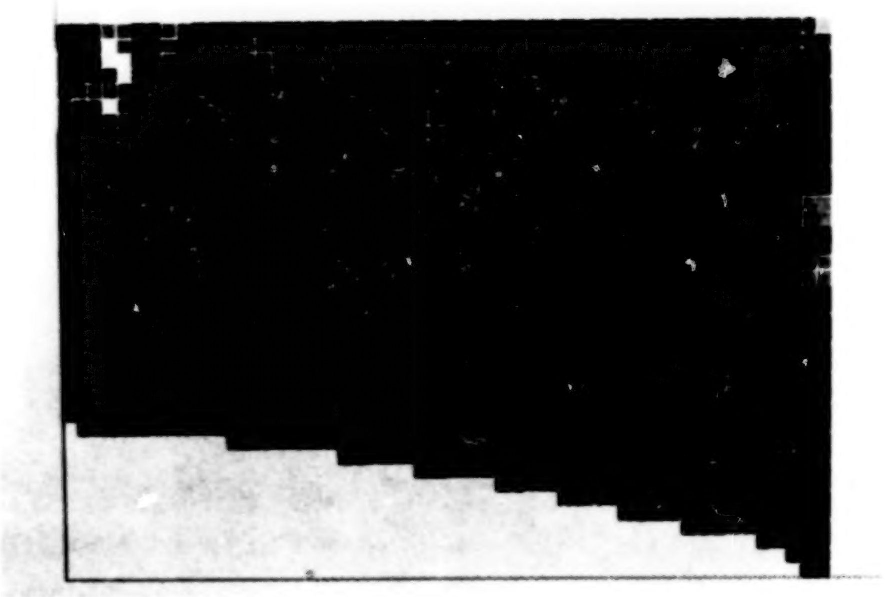


Figure 2. "Developed Land", 1977, SLUI test data.

alternative is less efficient than if the data were stored in their original form in flat files. A trade-off emerges between the benefits of the DBMS in distancing the user from application-specific formats and the controlled redundancy enforced by the relational structure. In this case it is not easy to judge qualitatively whether the advantage lies with user accessibility or with processing convenience and storage economy. The outcome depends on the application.

3.3 Planning and Management Data from Scotland (TRRU)

Data at a five (5) km scale of resolution were extracted for a study region 320 km by 100 km in the central lowlands of Scotland. Population data, land use categories, land form, relief, and transport network length were included for study. The first difficulty is the problem of handling continuous variables in the context of block encoding. TRRU data had not been previously clasified, so the user is faced with a choice between data loss or storing the data as individual cells with interval data in the attribute table.

In preparing the data it was determined that the population data had to be rounded off to the nearest thousand before any appreciable aggregation of individual cells into blocks was observed (Figure 4). As many cells have sparse population, this level of aggregation is unacceptable. The problem of

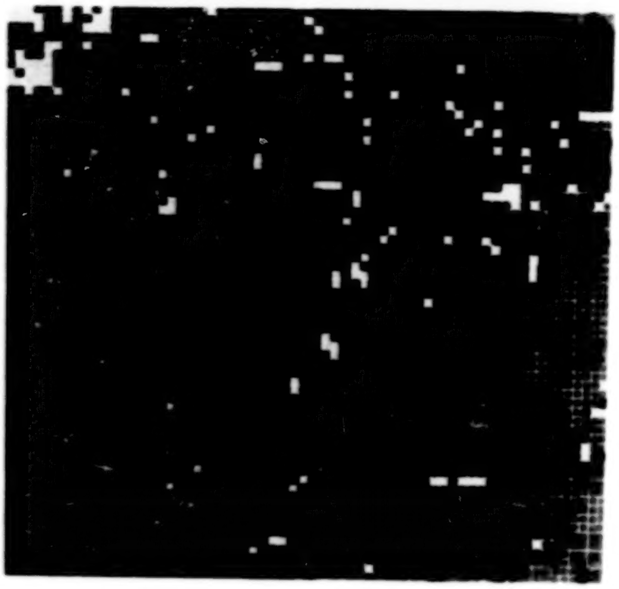


Figure 3. LANDSAT TM classification, Northern Belize.

aggregation is made worse by the highly irregular coastline of western Scotland and the resulting lack of contiguity. Assuming good reasons for collecting data on five km centers, the question of an appropriate level of precision is difficult to resolve conclusively for population and similar continuous variables. For the rail and road network data the problem is less acute. Network lengths were recorded to a resolution of plus or minus one meter. Loss of precision up to one km could be tolerated to facilitate a block decomposition. Given the relatively small number of cells in the grid coverage at this scale of resolution, the best approach is perhaps to retain the individual cell values.

3.4 Discussion

No formal benchmarking of processing times was attempted, but some generalizations are made, based on performance issues raised in the literature. In the quadtree block encoding algorithm, processing encoding time is proportional to the number of blocks to be encoded. In turn, the number of blocks to be encoded is related to image size, spatial resolution, and degree of within-image homogeneity. Where performance testing has been reported, the tendency is for 'average performance' to tend towards 'worst-case performance' (11, 12). As the time taken to decompose an image into blocks is proportional to the number of regions and inversely proportional to their average size, it then follows that performance behaves similarly since

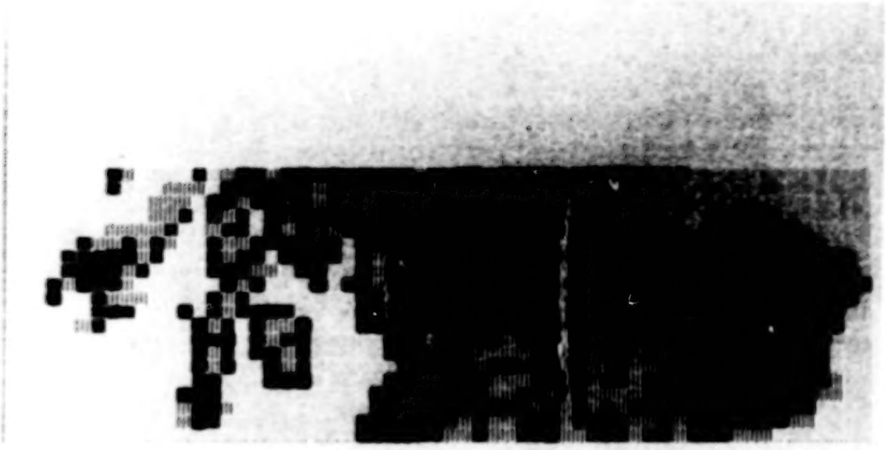


Figure 4. Population 1971, Central Scotland.

the ORACLE database searches and retrieves one row at a time even where multiple rows are returned in the cursor.

The retrieval of entity and attribute information for the geology image took over eight minutes CPU time on an unloaded DEC VAX 11-750 and four minutes on MICRO-VAX 2000. The efficiency of a GEOVIEW implementation of quadtrees entities is regulated by the need to make relational joins between the base entity, base attribute, and base attribute-symbolism tables at run time. A major drawback of the relational data structure is its slow speed across relational joins. The indexing of primary and foreign keys improved retrieval speeds by a factor of two. GEOVIEW performance is significantly improved by indexing the primary and foreign keys of the base entity and base attribute table respectively, and similarly between attribute and symbolism tables.

A second option is the physical rearrangement of tables on the storage medium. Frequently accessed blocks of stored data are clustered in close proximity to one another, thus reducing search and retrieval time. Clustering is not possible on tables containing a column of 'LONG' data type, therefore clustering is not possible on the GEOVIEW base entity table as designed. The inability to cluster tables is a major drawback, and the performance costs of that choice must be reckoned in the overall adjudication of the wisdom and success of the GEOVIEW design. Furthermore, in the particular quadtrees implementation described here, there are few direct uses of the LONG data field that do not compromise the integrity of the relational data structure. There is a case to be made for develoment of a modified base entity table without a LONG data type 'entity-data' column.

Removing the restriction on clustering may contribute to acceptable retrieval time for quadtree entities. Now the dilemma becomes one of forfeiting the simplicity of the GEOVIEW approach in favor of performance criteria. Tradeoffs between theory and performance are widespread in GIS and, given the need to produce results in reasonable (on-screen) time for very large data sets, performance often takes priority over theoretical purity. The cost of abandoning a strict posture towards database integrity is the loss of simplicity and elegance which characterize the GEOVIEW design. A transparent data structure can be easily understood and is more easy to develop, trouble-shoot, and maintain than an ad hoc design.

4. SUMMARY

The quadtree block is a geographical data entity that shows potential for the handling of certain types of geopraphic information, particularly for economy of storage space and for efficient search procedures and spatial operations. This study has demonstrated the quadtree block ordering of two-dimensional space and has raised theoretical questions about the best design of relational data structures to manage blocks in an integrated GIS.

The main conclusions from the study are that:

1) the GEOVIEW relational model can handle quadtree blocks as entities;

2) the GEOVIEW GIS is capable of retrieving and graphically displaying and manipulating the data and that given certain hardware and software improvements operational performance appears to be a realistic goal;

3) the demonstrated implementation of quadtree block entities without recourse to the LONG data type is expensive in storage terms; modified base entity tables could, however, provide major improvements in performance through the

clustering of related data on the disc;

4) as has been noted by others, including Samet (7), Mark (13), and Schaffer (8) certain textures and spatial distributions benefit, in storage and retrieval terms, more than others;

5) there is a need to distinguish between quadtree blocks as geographical entities and the entire quadtree structure as an entity, since both types have distinct value and can be implemented separately and concurrently in the GEOVIEW design. Exploration of GEOVIEW to handle quadtree structures is continuing.

REFERENCES

1. **Morehouse, S.**, 1987: "ARC/INFO: A Geo-Relational Model for Spatial Information." Proceedings, AutoCarto 8, Baltimore, MD, 338-397;

2. **Guptill, S. C.**, 1986: "A New Design for the U.S. Geologic Survey's National Digital Cartographic Database." Proceedings, AutoCarto London, M. Blakemore (ed), Vol.2: 519-528;

3. **van Roessel, J. W.**, 1987: "Design of a Spatial Data Structure Using the Relational Normal Forms." International Journal of Geographical Information Systems, 1, 1: 33-50;

4. **Waugh, T. C.**, 1987: "The GEOVIEW Design." Xerox Copy, Department of Geography, University of Edinburgh, Scotland;

5. **Waugh, T. C.**, and **Healey, R. G.**, 1987: "The GEOVIEW Design. A Relational Data Base Approach to Geographical Data Handling." International Journal of Geographical Information Systems, Vol. 1, 2: 101-118;

6. **Samet, H.**, 1984: "The Quadtree and Related Data Structures." ACM Computing Surveys, 16, 2: 87-260;

7. **Samet, H.**, 1985: "Reconstruction of Quadtrees from Quadtree Medial Axis Transforms." Computer Vision, Graphics and Image Processing, 29: 311-328;

8. **Schaffer, C. A.**, 1986: "Application of Alternative Quadtree Representations." CAR-TR-208, Center for Automation Research, University of Maryland, College Park, MD;

9. **TYDAC™ Technologies Inc.**, 1985: "Spatial Analysis System Summary (SPANS)." Promotional literature, Ottawa, Canada;

10. **Peuquet, D.**, 1984: "A Conceptual Framework and Comparison of Spatial Data Models." Cartographica, 21, 4;

11. **Dyer, C. R.**, 1982: "The Space Efficiency of Quadtrees." Computer Graphics and Image Processing, 19, 4: 335-348;

12. **Mark, D. M.**, and **Lauzon, J. P.**, 1985: "The Space Efficiency of Quadtrees: An Empirical Examination Including the Effects of Two-Dimensional Run-Encoding." Geo-Processing, 2: 367-383;

13. **Mark, D. M.**, 1986; "The Use of Quadtrees in Geographic Information Systems and Spatial Data Handling." Proceedings, AutoCarto, London, Vol. 1: 517-526;

BLANK PAGE

Object-Oriented Databases: Database Requirements for GIS*

Max J. Egenhofer
Andrew U. Frank
Surveying Engineering Program
University of Maine
Orono, ME 04469, USA
MAX@MECAN1.bitnet
FRANK@MECAN1.bitnet

Abstract

Conventional database management systems have proven to be insufficient to model and perform non-standard applications such as spatial information systems, CAD/CAM, etc. The object-oriented approach overcomes some crucial deficiencies with refined methods for data abstraction and suitable tools to structure data allowing the usage of complex object types. For the implementation of an object-oriented database management system appropriate tools in programming languages are needed. A Geographic Information System (GIS) which treats spatial data asks for additional object-oriented extensions such as storage clusters to provide fast access on spatial objects, concurrent management of spatial and non-spatial data, and treatment of properties of spatial objects in query languages.

1 Introduction

Geographic information systems (GIS) contain substantial amounts of data which must be stored in computer readable and accessible form. Computer scientists have studied database management systems (DBMS) for several years and database management systems for commercial usage are currently available; however, several studies have shown that these systems are not suitable for non-standard applications, such as GIS, Land Information Systems (LIS) [Frank 1981] [Frank 1984],

---

*Work on these concepts was partially funded by a grant from NSF under No. IST-8609123

and CAD/CAM [Eastman 1980] [Stonebraker 1982]. Several experimental database management systems have been tried, but a complete system is not yet available commercially and additional research is needed.

During the last years, research in software engineering has promoted an object-oriented design method by which real world objects and their relevant operations are modelled in a program. This approach is most useful for application areas like GIS, because it naturally supports treatment of complex, in this case geometric, objects [Kjerne 1986]. Unlike conventional data models, an object-oriented design is more flexible to describe the complex data structures, primarily by refined techniques for conceptual modeling, such as generalization and inheritance.

Spatial information systems will benefit from the use of object-oriented databases in various forms:

- The system architecture will become clearer and easier to maintain so that software systems will have a longer life cycle.
- Programmers need not worry about aspects of the physical location where to store data; instead, a unified set of commands provides the functionality to store and retrieve data.
- By using a database, data is treated by its properties; by using an object-oriented database, these properties are logically combined to objects which can be often very complex.

The paper starts with a review on existing database systems and their deficiencies in serving as a suitable tool for modeling space related applications. In particular, the theory for data structuring techniques is compared to the tools in programming languages available to implement them. Following this chapter, an object-oriented data model is introduced which is built upon the three major abstraction methods of classification, generalization, and aggregation. The important concepts of hierarchical and multiple inheritance are explained. Software engineering aspects are investigated in chapter 4. Tools, such as programming languages and programming environments, are seen as especially crucial to implement object-oriented designs. In chapter 5, properties and requirements of object-oriented database management systems suitable for GIS are discussed. The implications of object-oriented structures to query languages are sketched. The paper closes with a summarization of the GIS requirements for a database management system.

2 Databases for Non-Standard Applications

Databases have become an accepted tool for storing data in readable and accessible form, and systems for CAD/CAM or spatial information (Geographic or Land

BEST COPY AVAILABLE

Information Systems) integrate a database system into a larger software system [Frank 1983].

Unlike conventional systems, spatial information systems deal with data describing the real world, and they combine several domains which have not been studied sufficiently. Think of a Geographic Information System in which a large variety of diverse tasks, such as

- sophisticated treatment of real-world geometry
- measurements of different resolution and accuracy
- uncertain measurements and attribute classifications
- legal aspects
- management of time series of veriable attributes
- representation of data in different generalization levels

are combined in a single system. This combination of difficult areas may be a reason why spatial information systems have been so slow to come and only similar, but more restricted applications are used successfully [Tripp 1987] [Beckstedt 1987].

Spatial information systems can benefit from using database management systems because these provide a unique form of storing and accessing structured data. The problems of low-level data management are removed from the programmer's and end user's responsibility, and data can be descibed by their properties, not their physical structure.

Unfortunately, existing commercial database systems are not sufficient for applications to engineering tasks, often called non-standard applications, such as CAD/CAM for VLSI design or cartographic and geographic information systems. Each of these areas struggle with the same kind of problem: they contain substantial amounts of 'real world' data including geometric aspects which must be managed by a computer. Their composition is too complex to be managed in conventional database systems efficiently. Interactive systems require a certain degree of performance, independent of the size of the data sets—a demand which cannot be fulfilled by existing systems.

2.1 Deficiencies of Conventional Database Systems

Members of the CAD/CAM-community have complained that there are currently no database systems which are appropriate for their demands [Buchman 1985] [Sidle 1980] [Udagawa 1984]. They state that standard database systems do not fulfill requirements which are crucial for engineering applications:

- Performance is unacceptable when a database is populated with large amounts of data [Wilkins 1984] [Härder 1985] [Maier 1986]. Dealing with spatial data, this deficiency is particularly visible during interactive graphic sessions.

- The treatment of complex objects [Lorie 1983], such as molecules in chemistry [Batory 1984], spatial objects in GIS/LIS [Frank 1984], or circuits in VLSI is not supported. Spatial objects, for example, should not be forced to be artificially decomposed in smaller parts.

- Appropriate mechanisms for data structuring [Johnson 1983], especially for structuring real-world data, are missing.

Most conventional database management systems are built on the relational model [Codd 1982]. In the relational model, data are organized in tables or relations. Columns of the tables are called attributes and all values in an attribute are elements of a common domain; rows are records, tuples, or relation elements.

While this concept is suitable for modeling commercial data, such as bank accounting, it is too simplistic for modeling data describing the real world. Geographic data cannot be modelled as strings, reals, and integers in table format—the necessary data types are more complex. One property of complex object types is that an object type can be part of another (more) complex object type. Data types for two-dimensional coordinates, for example, consist of an x- and y-value. A more complex object type includes some error value stating the accuracy of the coordinates. Such 'error-affected' coordinate types may be part of a point type which combines the coordinates with a number and a theme.

Moreover, the relational model lacks the powerful concept of recursion which is crucial for modeling complex situations such as spatial data and their subdivisions: areas can be decomposed into several sub-parts which themselves can be continuously decomposed further. For example, the area of a town is composed of several other areas, such as the house lots, streets, etc.

Only recently, the design of databases for spatial information system has become a research topic [Lipeck 1986] [Schek 1986] and only a few experimental database systems exist which pursue new concepts [Frank 1982a] [Dayal 1986] [Batory 1986].

In existing GIS software systems, a trend towards sophisticated software engineering methods [Aronson 1983] and architecture [Smith 1986] can be observed, stressing object-oriented concepts for geometric data handling [Herring 1987]; however, database management systems, and especially object-oriented ones, have not yet been incorporated into commercial GIS systems.

What are the technical reasons that database systems have failed to support complex non-standard situations? First, database requirements are not a hardware problem. It is commonly known that hardware currently develops much faster than software; however, faster and less expensive CPUs, larger hard disks, and more memory do not overcome certain problems in database technology for engineering. Solutions which can be achieved by exploiting additional and faster hardware are not the problems which will be addressed in this paper. For example, new technologies for hard disks provide more storage capacity at less expensive costs, but the disk access has not become faster. This is important for database management systems which struggle with growing data collections which get too complex to be managed efficiently.

Rather than hardware, software engineering is an impediment for better-suited engineering databases. We claim that theoretical and conceptual problems are the subject of major improvements. For example, even with much faster access to storage devices, internal data structures organizing spatial data will be needed in order to provide adequate performance for range queries.

2.2 Deficiencies of Software Engineering Tools

Programming is still seen as an art in which the programmer can accomplish a goal by whatever means he believes to be suitable. The contrary is true: each program is a very formal piece of code following strict rules; programs must not only be compatible between hardware, but also 'compatible' among programmers, i.e., different programmers must be able to read and understand each others' code. By following strict rules and restrictions, these goals can be achieved. Programming languages tend towards offering a large variety of tools, while the opposite is needed, namely restrictions, such that several programmers come up with very similar solutions.

The other deficiency observed is that concepts in software engineering do not match with database models or are not suitable for them. Theory in software engineering provided some suitable techniques, such as the concept of abstract data types [Guttag 1977] [Parnas 1978] [Zilles 1984], in which each module encapsulates an object type with all its pertinent operations. Standard database systems, based on networks [CODASYL 1971] or the relational [Codd 1970] data model do not fit easily into this concept.

2.3 Deficiencies of Implementing Data Structures

A number of methods to capture more semantics in the data model have been proposed in the literature (for an overview and critique see [Brodie 1984] which contains an extensive list of references), but most of these methods have not been implemented, and none of them is readily available. Complex tasks require complex structuring tools. 'Complex', however, need not mean 'complicated': the support of data structuring must be powerful without sacrificing the ease of use, and it must be extensive without becoming excessive.

In the past, considerable efforts were made to enrich existing data models with facilities to treat complex objects. 'QUEL as a Datatype' [Stonebraker 1983a] and ADT-INGRES [Stonebraker 1983] extend the relational model with features to define more complex types; DAPLEX [Shipman 1981] is a functional language which includes hierarchical relationships and transitive closure; the NF^2 model [Schek 1985] supports composite attribute types being tuples or relations performing a hybrid of relational and hierarchical data model. Recently, it was investigated, how geometry could be modelled by using these extended data models [Kemper 1987], and it was shown that only partial remedies are provided with these extensions.

3 An Object-Oriented Data Model

This chapter introduces the notation of objects and the abstraction tools. An entity of whatever complexity and structure can be represented by exactly one object in the database [Dittrich 1986], meaning that no artificial decomposition into simpler parts should be necessary due to some technical restrictions. Complex data types for large objects, such as an entire city (with its details about streets, houses, and their details such as owners, neighbors, etc.), do not overcome the problem of data structuring.

The object-oriented data model is built on the three basic concepts of abstraction [Brodie 1984a]: classification, generalization, and aggregation. Furthermore, inheritance describes how the properties of a class are derived from properties of related classes.

3.1 Classification

Classification can be expressed as the mapping of several objects (instances) to a common class. The word *object* is used for a single occurrence (instantiation) of data describing something that has some individuality and some observable behavior. The terms *object type*, *sort*, *type*, *abstract data type*, or *module* refer to types of objects, depending on the context. In the object-oriented approach, every object is an instance of a class. A type characterizes the behaviour of its instances by describing the operators that can manipulate those objects [O'Brien 1986]. These operations are the only means to manipulate objects. All objects that belong to the same class are described by the same properties and have the same operations.

For example, the model for a *town* may include the classes *residence*, *commercial building*, *street*, *park*, etc. A single instance, such as the house with the address '30 Grove Street', is an object of the corresponding object type, i.e. the particular object is an instance of the class *residence*. Operations and properties are assigned to object types, so for instance the class *residence* may have the property *number of bedrooms* which is specific for all residences. Similarly, the class *street* may have an operation to determine all adjacent parks.

In the implementation, classes translate to abstract data types or modules.

3.2 Generalization

Generalization is a well-known term in cartography for varying representations of objects according to the level of detail needed in a given scale. Similarly, generalization as an abstraction mechanism provides views in different levels of detail.

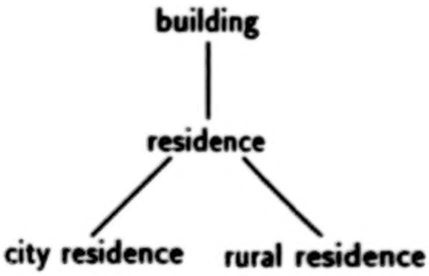
Several classes of objects which have some operations in common are grouped together into a more general *superclass* [Dahl 1968] [Goldberg 1983]. The converse relation of superclass, the *subclass*, describes a specialization of the superclass. *Ancestor* and *descendant* are often used as equivalent terms for superclass and subclass. Subclass and superclass are related by an *is_a* relation. For example, the object type *residence* is a *building*; *residence* is a subclass of *building*, while *building* is its superclass.

3.2.1 Inheritance

The properties and methods of the subclasses depend upon the structure and properties of the superclass(es). Inheritance defines a class in terms of one or more other classes. Properties which are common for superclass and subclasses are defined only once (with the superclass), and inherited by all objects of the subclass, but subclasses can have additional, specific properties and operations which are not shared from the superclass. Inheritance is transitive propagating the properties from one superclass to all related subclasses, to their subclasses, etc. This concept is very powerful, because it reduces information redundancy [Woelk 1987]. For example, *building* shares properties and operations with *residence*, such as having an *area* and being *neighbor* of some other building. The class *residence* inherits all operations from its superclass *building* without the need to redefine them explicitly. Additional properties, such as the *number of bedrooms*, are specific for the *residence*. Specializations of *residences*, such as *rural residences* and *city residences* inherit the specific properties of the *residences*, i.e. *number of bedrooms*, and by transitivity the properties of the super-superclass *building*, i.e., *area* and *neighbor*.

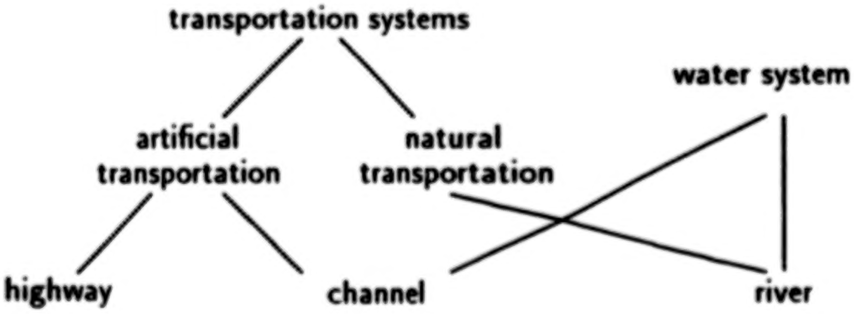
Operations of the superclass are compatible between objects of the superclass and subclass. Every operation on an object of a superclass can be carried out on the subclass as well; however, operations specifically defined for the subclass are not compatible with superclass objects. For example, *neighbor* is an operation of the superclass *building*, and it is thus compatible with objects of the type *residence*. On the other hand, the operation *number of bedrooms* is specific fo the subtype *residence* and thus not applicable for objects of the superclass *building*.

Hierarchies: Inheritance can be strictly hierarchical as in the example of *building*, *residences*, etc. Hierarchical inheritance implies that each subclass belongs only to a single group of hierarchies; one class cannot be part of several distinct hierarchies, i.e., in hierarchical generalization, a class can have only one direct superclass.



Multiple Inheritance: The structure of a strict hierarchy is an idealized model and fails most often when applied to real world data. Most 'hierarchies' have a few non-hierarchical exceptions in which one subclass has more than a single, direct superclass. Thus, pure hierarchies are not always the adequate structure for inheritance; instead, class lattices [Woelk 1987] with acyclic directed graphs are more suitable. This concept, allowing a multitude of distinct superclasses for a single class, is called multiple inheritance [Nguyen 1986].

For example, *highways* and *channels* are *artificial transportation ways*, while *rivers* are *natural transportation systems*; however, *channels* and *rivers* both belong to another hierarchy, the *water system* as well. So, *channel* and *river* participate in both *water systems* and *transportation ways*, two distinct hierarchies which cannot be compared with each other.



3.3 Aggregation

Several objects can be combined to form a semantically higher-level object where each part has its own functionality. This is different from the way generalization is defined: operations of aggregates are not compatible with operations on parts.

BEST COPY AVAILABLE

Aggregation establishes a relation which is often called a 'part-of'-relation since aggregated classes are 'parts of' the aggregate. For example, the class *county* is an aggregate of all related *settlements*, *forests*, *lakes*, *streets*, etc.

3.3.1 Dependencies Among Values of Different Object Types

Complex objects often do not own independent data, but properties which rely upon values of other objects. In GIS and LIS, for example, a large amount of attribute values is propagated from one level of abstraction to another. When combining local and regional data, this concept must be used to pursue the dependencies among data of different levels of resolution [Egenhofer 1986]. The population of a county, for example, is the sum of the population of all related settlements; therefore, the property *population* of the aggregate *county* is derived by adding all values of the property *population* owned by the class *settlement*.

While inheritance is the propagation of operations, functional dependencies describe how *values* of one class are derived from values of another class. Often, a value is directly propagated from the property of one class to another property of a different class. In the object-oriented model, objects may have properties with values which rely on values of other objects. Dependent values must be derived.

If more than a single value contributes to the derived value, the combination of the values must be described by a function. Common operations are minimum, maximum, sum, average, and weighted average.

3.4 Tools for Modeling

Even for relatively small models of a mini-world, the structure can become too confusing to be presented with pure alphanumeric means. Graphical methods have proven to be suitable tools for a better and clearer understanding of data structures; representations such as entity-relationship diagrams [Chen 1976] are essential tools in an object-oriented design. By using the power of graphical representation, modeling becomes more clear and understandable compared to pure alphanumerical descriptions.

The concept of multiple inheritance must be incorporated into the entity-relationship model by using some simple graphical means.

4 Software Engineering Aspects

A database is only one part in the large effort to produce a CAD/CAM system, a land information system, or any of the other non-standard applications. It is thus necessary that database methods and techniques fit well within a software

BEST COPY AVAILABLE

engineering environment. Theoretically founded methods have proven to be well-suited for modularization based on abstract data types.

4.1 Abstract Data Types

The software engineering method for abstraction and data structuring is based on the work on formal specifications using abstract data types. An abstract data type (ADT) or a multi-sorted algebra is a mathematical structure which fully defines the behaviour of objects, i.e., the semantics of the object-type and its operations. Abstract data types are specified as an algebra describing what sorts of objects (types) are dealt with and what kinds of operations they are subject to. A set of axioms determines the effects of the operations. It is up to the designer to assure that the sorts and their operations form a reasonable and meaningful object.

Abstract data types can be combined in layers, where higher-level abstract data types are first described independently (specifications) and it is then shown how this behaviour can be achieved using other, hierarchically lower abstract data types (called an 'abstract implementation' [Olthoff 1985] [Frank 1986]). Such abstract implementations can be formally checked for correctness, by proving that the axiomatically specified behaviour of the upper abstract data type follows from the abstract implementation and the axiomatically specified behaviour of the lower abstract data type.

Abstract data types are very useful in the design of large systems. They allow a much more comprehensive and complete way of specifying routines together with their axioms than an extensive programming language does.

4.2 Object-Oriented Programming

The programming language must support the abstraction methods. The development of object-oriented databases has had problems, mostly because suitable software tools are missing or old and inappropriate tools have been used. Certain programming tools must be available to implement object-oriented databases and data structures.

- In order to implement complex objects, language tools, such as RECORD structures in Pascal [Jensen 1978], Ada [Ada 1983], etc., are needed to create user-defined object types. For example, in a spatial information system a large amount of the data are points with coordinates, either 2-dimensional or 3-dimensional. By combining the coordinates to a RECORD, a complex data type, such as a `pointType`, can be created.

```
TYPE pointType = RECORD
                    x, y, z: coordType;
                 END;
```

- In object-oriented programming, objects consist of a type definition and a collection of operations. Objects can be only accessed or manipulated by using these operations. For example, the object type `pointType`, as introduced above, has a specific operation to shift a point in x-, y-, or z-direction (translation); the user can access a point only via the interface of the `pointType`.

  Object-oriented programming needs (1) tools to tie types and their operations closely together (modularisation), and (2) methods to hide internal parts of the routines from unauthorized external usage (encapsulation). Modula-2 [Wirth 1982], Smalltalk-80 [Goldberg 1983], and C++ [Stroustrup 1986] are examples for languages which support modularization and encapsulation. Encapsulation provides implementation-independent code outside of a module.

- Genericity [Cardelli 1985] has proven to be a suitable and powerful method to reduce redundant definitions of ADTs. A generic type or object is a definition which is a backbone for a series of detailed and specified definitions. A *pair*, for instance, is a generic object type which combines two objects; *pair* can be applied to two integers forming a *pair of integers*. Operations which are common for all *pairs*, such as *equality*, are defined only once, while their implementation might vary from type to type.

  Programming code is considerably reduced by using generic types, because operations which apply to all instances are coded only once.

- Only recently, inheritance has been recognized as a crucial issue in object-oriented approaches. Two categories of inheritance are identified, hierarchical (or straight) and multiple inheritance.

  Hierarchical inheritance deals with a strict structure in which for each child only a single direct ancestor exists. Some programming languages, such as Smalltalk-80 and C++, support straight inheritance, other languages with variant RECORD structures, such as Pascal or Ada, can at least simulate hierarchical inheritance of types, but not of operations. For example, two-dimensional and three-dimensional coordinates are both representations for points. From this superclass, they both inherit operations such as calculating the distance between two points. The implementation of these operations may be different, however, from outside they look the same. With a variant RECORD structure, the inheritance can be implemented as follows:

BEST COPY
AVAILABLE

```
TYPE pointType = RECORD
                   CASE dimension: dimensionType of
                      twoD: (x, y: coordType);
                      threeD: (x, y, z: coordType);
                 END;
```

Multiple inheritance allows one object to be part of several distinct hierarchies. Only a few languages, such as some LISP dialects (CommonObjects [Snyder 1986], Zeta Lisp [Weinreb 1981]), ObjectLOGO [Davidson 1987], and Trellis/Owl [O'Brien 1986] allow the definition of multiple inheritance.

These concepts for object-oriented programming do not include a message-passing paradigm [Goldberg 1983]; message-passing is based on passive objects which can receive messages directing what actions can be executed on an object. Message-passing is often cited as necessary for an object-oriented design; in our opinion it is not an essential feature, but certainly a desirable and helpful paradigm. It was outlined that message-passing is rather a pedagogical than a semantical difference to conventional routine calls, and any procedure call in an Algol-like language could be seen as message-passing [Storm 1986].

Object-oriented programming of object-oriented applications depends on the choice of the programming language; only a few languages support object-orientation sufficiently. We claim that object-oriented applications cannot be achieved without object-oriented tools and concepts.

4.3 Support System

Database systems are large software systems which must be embedded in some environment supporting development and maintainance. Especially the latter is crucial for the life cycle of a system. Such an environment must be tailored to the object-oriented approach by automatically propagating object definitions to other objects, controlling which object uses which other objects, and which objects are used elsewhere.

In software engineering, abstract data types may be represented by modules which encapsulate a type and its operations. Facilities such as packages and use clauses in Ada or modules and import commands in Modula-2 allow the programmer to easily translate abstract data types into executable code. For other languages with capabilities for separate compilation of modules, these concepts can be applied by using a precompiler which propagates the ADT-definitions to the modules where they are used.

5 Database Management Systems Tailored to GIS

This chapter investigates how the object-orientation reflects in the architecture of a database management system for GIS. Ideally, an object-oriented database consists of subsystems which can be added or exchanged to support specific tasks [Batory 1986]. Before investigating the GIS-specific parts of a tailored object-oriented database, subsystems of standard database systems are compiled which can be adopted from an object-oriented database.

5.1 Applicable Subsystems of a Conventional Database

5.1.1 Disk Storage Subsystem

A database must provide persistent storage for objects modelled in the information system. Due to the large size of the data sets, use of magnetic or optical disk systems for permanent storage is required.

A storage subsystem must provide operations to store and retrieve data. The storage subsystem does not know anything about other operations owned by the objects or their internal structure. Disk access is expensive and improvements in performance can be achieved by reducing the number of disk accesses by buffering the storage elements.

5.1.2 Multi-User Facilities

In multi-purpose information systems, a database is shared by a large variety of users who often want to access the same data in parallel. The known straight forward techniques are sufficient as long as concurrent users only access the data without changing them. This is, however, usually not appropriate to accomodate users and their needs. Most organizations cannot restrict update operations to a single user without severe distortion of their flow of work.

A database management system must provide control mechanisms in order to guarantee concurrent access for multiple users to all data and to prevent users from accessing data during inconsistent states, i.e., the database management system guarantees that no user sees the preparative stages of a change until the change is completed and made visible to all other users. This is done by a dedicated subsystem, the so-called transaction manager. A transaction is a sequence of operations by one user which transforms the database from one consistent state into another one, such that outside of a transaction a consistent image of the mini-world exists at all times. A transaction can either be committed, i.e., all modifications will be made visible, or aborted, i.e., none of the modifications started during the transaction will become effective. Modified data are isolated during the transaction preventing other users from uncontrolled access. The transaction subsystem can fulfill with essentially the same mechanisms other goals as well. It is usually assumed that a transaction system will maintain all the consistency of the database across hardware or software malfunctionings, and prevents loss of data. A transaction has the following four classical properties (ACID) [Härder 1985]:

- Atomicity states that in a transaction either all modifications become effective, or no change at all are made in the data set.
- Consistency is guaranteed before and after the transaction.
- Isolation preventes from unauthorized access during a transaction.
- Durability keeps comitted transactions permanently.

5.1.3 Distributed Systems

Central databases are important resources and access to them must be available for many operations in parallel and at the same time. For organizational reasons, large databases are often not centralized, but distributed over several computer systems. This may improve their availability during local hardware failures, reduce data transfer cost, etc. In our opinion, support for distribution should be built into this layer of the subsystem.

5.2 Specific GIS requirements

In this chapter it is proposed which additional concepts must be integrated into an object-oriented database management system, such that it would fulfill the high expectations and requirements on a persistent management system for very large spatial data sets.

5.2.1 Performance Enhancements for Spatial Data

Large spatial data collections should be managed without artificially subdividing them into several user-visible map sheets. Ideally, the user should have a single database covering the entire area of his or her application; however, the performance penalties are serious when very large amounts of spatial data should be managed. Interactive graphic representations of spatial data with map output require especially adequate performance.

Generally, database performance depends upon the number of disk accesses needed to retrieve the data. Conventional systems do not pay attention to the distribution of spatial data and store them 'as they are collected'. This treatment is not appropriate for spatial data collections, because the access on a set of data to be drawn as a map will take a long time due to too many disk accesses. The larger the data collection grows, the slower the system will perform.

Access on spatial data must be enhanced by integrating spatial storage clusters [Frank 1983a]. Such techniques take advantage of the spatial distribution of data and store them such that spatial neighbors are neighbors on the storage device. It is assumed that (1) data used on one drawing are likely to be re-used soon for further display or interactive modification, and (2) not only a single object, but several adjacent objects will be used on a drawing. Spatial access is organized such that not a single instance, but a set of adjacent data is read at a time from a storage device and afterwards kept in main memory for future access. This buffering scheme needs to be organized according to some rules, e.g., a least-frequently-used strategy. In combination with spatial access methods, fast response time can be guaranteed essentially independent from the size of the data collection.

It is worthwile to study how storage structures for heterogeneously distributed spatial data can be improved such that access on less frequent objects performs well [Egenhofer 1987].

5.2.2 Complex Geographical Objects

The complexity of geographical objects, primarily spatial objects, requires methods to define and use appropriate data types and operations. The object-oriented model is tailored to this task. Data structures for recursive object definitions, such as areas being subdivided in other areas, and transitive closure operations are necessary.

Geometric and non-spatial data should not be forced to be physically separated from each other or managed in different types of databases. Instead, both categories must be integrated in one single system. This requires that

- user-defined, complex types can be stored in the database.
- a buffering schema is incorporated to reduce physical disk access; the geometric neighborhood should be exploited for a system with primarily graphical output.

Since objects can be more complex, their transactions will last longer and may be nested. A transaction concept is necessary which goes beyond the classical properties atomicity, consistency, isolation, and durability [Härder 1985].

5.3 Object-Orientation in Query Languages

Query languages have been disregarded for a long time, while they are one of the most crucial issues of an object-oriented database. Query languages are expected to benefit from the object-oriented approach by providing object operations at the user interface. Standard query languages for conventional database mangement systems, such as SQL [Chamberlin 1976], QUEL [Stonebraker 1976], or Query-By-Example [Zloof 1977], are not suited to deal with spatial data, because they do not include the specific properties of spatial objects. Non-standard database management systems must be furnished with query languages which support the treatment of complex objects including their specific properties. Proposals have been made to develop completely new query languages which are tailored to their specific application [Frank 1982]. A database system for GIS which is able to manage spatial objects efficiently is incomplete if it does not support spatial data and their properties in the query language.

We do not believe that natural language interfaces overcome the problems in query languages for non-standard applications and rather promote structured, object-oriented query languages.

5.3.1 Support of Spatial Relations

In order to treat spatial objects in query languages, properties must be included which are specific for spatial data. Typical properties among spatial objects are topological and metrical relations describing neighborhood, inclusion, distance, and direction. Object-oriented spatial relations must be dimension-independent, i.e., such that they can be applied to any spatial object. 'Disjoint', for example, is a relation which can hold for any two spatial objects (two points, two lines, two areas, two volumes, but also a point and a line, a point and an area, etc.).

Currently, we are investigating how the syntax of a standard query language must be extended such that it includes an adequate treatment of spatial objects [Egenhofer 1987a].

5.3.2 High-Level Object-Oriented Operators

Conventional query languages address subparts of objects explicitly and compare them with standardized operations. This is the reason for the 'artificial' style of queries such as

```
RETRIEVE all roads
WITH road.width > 12
```

In natural language, humans would combine the operation ¿ with the type of the object-subpart to some meaningfull term such as *broader than*. The object-oriented approach pursues this concept since the structure of the object and its operations are closely combined. Consequently, query languages must apply this concept, too, and express relations between objects such as

```
RETRIEVE all roads
WHICH ARE broader than 12
```

By combining object parts with operators and mapping them onto high-level operators, an object-oriented view of operators can be presented.

5.3.3 Graphical Display

Conventional query languages deal only with alphanumerical data, consequently, the graphical display for spatial objects cannot be specified. The specification is more complex for graphical output than for alphanumerical output where the sequence of columns in a table is described. Methods are needed to specify colors, patterns, and symbols.

Furthermore, interactive sessions with graphical output need mechanisms for user feedback, such as input via mouse on a drawing. The concept of direct manipulation on objects is more advanced and object-oriented than conventional input methods, because it does not reference a specific value, but screen coordinates which correspond to some object.

The output system directed by a query language has some other interesting problems, such as the selection of context [Frank 1982] which is dispalyed to make the graphical output understandable. Other proposals investigate, how methods from artificial intelligence like LOBSTER [Frank 1984a] can be incorporated into an intelligent interface for query languages.

6 Conclusion

It has been investigated how object-oriented database management systems can serve as suitable tools for spatial information systems. It was outlined that current database technology is not sufficient for the specific tasks when dealing with large amounts of spatial data. Recently, research in non-standard database environments promoted an object-oriented model which looks promissing to overcome some problems that make conventional database management systems unsuitable, such as the lack of modeling power to adequatly describe complex objects and the unacceptably slow performance of current implementations. This paper presented an object-oriented data model based on the abstraction concepts of classification, generalization, and aggregation. Hierarchical and multiple inheritance are crucial for modeling complex object types. The implementation of object-oriented data structures was investigated from the software engineering aspects (object-oriented programming languages).

With respect to geographic applications, several components of an object-oriented database system were identified which are mandatory for treating very large collections of spatial and non-spatial data. In particular, data structures for complex spatial data types, internal storage clusters for fast access on spatial data, and concurrent treatment of spatial and non-spatial objects must be incorporated in the database management system. Finally, the impact of the object-oriented design on query languages was investigated. Query languages for spatial information systems need high-level relations and operators for spatial objects, methods for specifying graphical display, and language tools for direct manipulation.

Acknowledgement

Thanks to Doug Hudson and Jeff Jackson who helped prepare this paper.

References

[Ada 1983] United States Department of Defense. Reference Manual of the ADA Programming Language. Springer Verlag, New York (NY), 1983.

[Aronson 1983] P. Aronson and S. Morehouse. The ARC/INFO Map Library: A Design for a Digital Geographic Database. In: Auto-Carto VI, 1983.

[Batory 1986] D.S. Batory. GENESIS: A Project to Develop an Extensible Database Management System. In: International Workshop in Object-Oriented Database Systems, Pacific Grove (CA), 1986.

[Batory 1984] D.S. Batory and A.P. Buchmann. Molecular Objects, Abstract Data Types, and Data Models: A Framework. In: 10th VLDB conference, Singapore, 1984.

[Beckstedt 1987] M. Beckstedt. Clark County Land Parcel Mapping System Pilot Study. In: Proceedings AM/FM International Automated Mapping/Facilities Management Conference X, Snowmass (CO), 1987.

[Brodie 1984] M.L. Brodie et al., editors. On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages. Springer Verlag, New York (NY), 1984.

[Brodie 1984a] M.L. Brodie. On the development of data models. In: M.L. Brodie et al., editors, On Conceptual Modelling, Perspectives from Artificial

Intelligence, Databases, and Programming Languages, Springer Verlag, New York (NY), 1984.

[Buchman 1985] A.P. Buchmann and C.P. de Celis. An Architecture and Data Model for CAD Databases. In: 11th VLDB conference, Stockholm, 1985.

[Cardelli 1985] L. Cardelli and P. Wegner. On Understanding Type, Data Abstraction, and Polymorphism. ACM Computing Surveys, 17(4), 1985.

[Chamberlin 1976] D.D. Chamberlin et al. SEQUEL 2: A Unified Approach to Data Definiton, Manipulation, and Control. IBM Journal of Research and Development 20(6), 1976.

[Chen 1976] P.S.S. Chen. The Entity-Relationship model: towards a unified view of data. ACM Transactions on Database systems, 1(1), March 1976.

[CODASYL 1971] Data Base Task Group report. CODASYL. Technical Report, New York (NY), April 1971.

[Codd 1982] E.F. Codd. Relational data base: a practical foundation for productivity. Communications of the ACM, 25(2), February 1982.

[Codd 1970] E.F. Codd. A relational Model for large shared data banks. Communications of the ACM, 13(6), June 1970.

[Dahl 1968] O.-J. Dahl et al. SIMULA 67 Common Base Language. Technical Report, Norwegian Computing Center, Oslo, 1968.

[Davidson 1987] L.J. Davidson et al. ObjectLogo. Coral Software Corporation, Cambridge (MA), 1987.

[Dayal 1986] U. Dayal and J.M. Smith. PROBE: A Knowledge-Oriented Database Management System. In: M.L. Brodie and J. Mylopoulos, editors, On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies, Springer Verlag, New York (NY), 1986.

[Dittrich 1986] K. Dittrich. Object-Oriented Systems: the Notation and the Issues. In: International Workshop in Object-Oriented Database Systems, Pacific Grove (CA), 1986.

[Eastman 1980] C. Eastman. System facilities for CAD databases. In: 17th Design Automation Conference, Minneapolis, 1980.

[Egenhofer 1987] M. Egenhofer. Managing Spatial Storage For Structured Data. In: Congres Conjoint Carto-Québec / A.C.C. / C.C.A., Québec, 1987.

[Egenhofer 1987a] M. Egenhofer. An Extended SQL Syntax For Treating Spatial Objects. In: Y.C. Lee, editor, Proceedings of the Second International Seminar on Trends and Concerns of Spatial Sciencies, Fredericton (NB), 1987.

[Egenhofer 1986] M. Egenhofer and A. Frank. Connection between local and regional: additional 'intelligence' needed. In: FIG XVIII. International Congress of Surveyors, Commission 3, Land Information Systems, Toronto, 1986.

[Frank 1986] A. Frank and W. Kuhn. A Provable Correct Method for the Storage of Geometry. In: Second International Symposium On Spatial Data Handling, Seattle (WA), 1986.

[Frank 1984] A. Frank. Requirements for Database Systems Suitable to Manage Large Spatial Databases. In: International Symposium On Spatial Data Handling, Zurich (Switzerland), 1984.

[Frank 1984a] A. Frank. Extending a Database with Prolog. In: L. Kerschberg, editor, Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island (SC), October 1984.

[Frank 1983] A. Frank. Data Structures for Land Information Systems—Semantical, Topological, and Spatial Relations in Data of Geo-Sciences. PhD thesis, Swiss Federal Institute of Technology, Zurich (Switzerland), 1983.

[Frank 1983a] A. Frank. Problems of Realizing LIS: Storage Methods for Space Related Data: The Field Tree. Technical Report 71, Swiss Federal Institute of Technology, Zurich (Switzerland), 1983.

[Frank 1982] A. Frank. MAPQUERY—Database Query Language for Retrieval of Geometric Data and its Graphical Representation. In: D. Bergeron, editor, Proceedings SIGGRAPH '82, ACM Computer Graphics, July 1982.

[Frank 1982a] A. Frank. PANDA—A Pascal Network Database System. In: G.W. Gorsline, editor, Proceedings of the Fifth Symposium on Small System, Colorado Springs (CO), 1982.

[Frank 1981] A. Frank. Applications of DBMS to Land Information Systems. In: C. Zaniolo and C. Delobel, editors, Seventh International Conference on Very Large Data Bases, Cannes (France), 1981.

[Goldberg 1983] A. Goldberg and D. Robson. Smalltalk-80. Addison-Wesley Publishing Company, 1983.

[Guttag 1977] J. Guttag. Abstract Data Types And The Development Of Data Structures. Communications of the ACM, June 1977.

[Härder 1985] T. Härder and A. Reuter. Architecture of Database Systems for Non-Standard Applications (in German),. In: A. Blaser and P. Pistor, editors, Database Systems in Office, Engineering, and Scientific Environment, Springer Verlag, New York (NJ), 1985.

[Herring 1987] J. Herring. TIGRIS: Topologically Integrated Geographic Information Systems. In: N.R. Chrisman, editor, Eighth International Symposium on Computer-Assisted Cartography, Baltimore (MD), 1987.

[Jensen 1978] K. Jensen and N. Wirth. Pascal User Manual and Report. Springer-Verlag, New York (NY), 1978.

[Johnson 1983] H.R. Johnson et al. A DBMS Facility for Handling Structured Engineering Entities. In: ACM Engineering Design Applications, 1983.

[Kemper 1987] A. Kemper and M. Wallrath. An Analysis of Geometric Modeling in Database Systems. ACM Computing Surveys, 19(1), March 1987.

[Kjerne 1986] D. Kjerne and K.J. Dueker. Modeling Cadastral Spatial Relationships Using an Object-Oriented Language. In: Second International Symposium on Spatial Data Handling, Seattle (WA), 1986.

[Lipeck 1986] U.W. Lipeck and K. Neumann. Modelling and Maipulation of Objects in Geoscientific Databases. In: 5th International Conference on the Entity-Relationship Approach, Dijon, 1986.

[Lorie 1983] R. Lorie and W. Plouffe. Complex Objects and Their Use in Design Transactions. In: ACM Engineering Design Applications, 1983.

[Maier 1986] D. Maier. Why Object-Oriented Databases Can Succeed Where Others Have Failed. In: International Workshop in Object-Oriented Database Systems, Pacific Grove (CA), 1986.

[Nguyen 1986] V. Nguyen and B. Hailpern. A generalized Object Model. SIGPLAN Notices, 21(10), October 1986.

[O'Brien 1986] P. O'Brien et al. Persistent and Shared Objects in Trellis/Owl. In: International Workshop in Object-Oriented Database Systems, Pacific Grove (CA), 1986.

[Olthoff 1985] W. Olthoff. An Overview on ModPascal. SIGPLAN Notices, 20(10), October 1985.

[Parnas 1978] D.L. Parnas and J.E. Share. Language facilities for supporting the use of data abstraction in the development of software systems. Technical Report, Naval Reasearch Laboratory, Washington (DC), 1978.

[Schek 1986] H.-J. Schek and W. Waterfeld. A Database Kernel System for Geoscientific Applications. In: Second International Symposium on Spatial Data Handling, Seattle (WA), 1986.

[Schek 1985] H.-J. Schek. Towards a Basic Relationsal NF^2 Algebra Processor. In: Conference on Found. Data Org. (FODO), Kyoto, 1985.

[Shipman 1981] D. Shipman. The functional data model and the data language DAPLEX. ACM Transactions on Databse Systems, 6(1), March 1981.

[Sidle 1980] T.W. Sidle. Weakness of Commercial Database Management Systems in Engineering Applications. In: 17th Design Automation Conference, 1980.

[Smith 1986] T. Smith et al. KBGIS-II: A Knowledge-Based Geographic Information System. Technical Report TRCS86-13, Dept. of Computer Science, University of California, Santa Barbara (CA), May 1986.

[Snyder 1986] A. Snyder. CommonObjects: An Overview. SIGPLAN Notices, 21(10), October 1986.

[Stonebraker 1983] M. Stonebraker et al. Application of abstract data types and abstract indices to CAD databases. In: Proceedings of ACM SIGMOD Conference on Engineering Design Applications, San Jose (CA), 1983.

[Stonebraker 1983a] M. Stonebraker et al. QUEL as a datatype. Technical Report UCB/ERL M83/73, University of California, Berkley (CA), 1983.

[Stonebraker 1982] M. Stonebraker and A. Guttman. Using a Relational Database Management System for CAD data. IEEE Database Engineering, 5(2), June 1982.

[Stonebraker 1976] M. Stonebraker et al. The design and implementation of INGRES. ACM Transactions on Database systems, 1(3), September 1976.

[Storm 1986] R. Storm. A comparison of the Objet-Oriented and Process Paradigms. SIGPLAN Notices, 21(10), October 1986.

[Stroustrup 1986] B. Stroustrup. The C++ Programming Language. Addison-Wesley Publishing Company, 1986.

[Tripp 1987] R.E. Tripp. AM/FM Progress at Carolina Power & Light Company. In: Proceedings AM/FM International Automated Mapping/Facilities Management Conference X, Snowmass (CO), 1987.

[Udagawa 1984] Y. Udagawa and T. Mizoguchi. An Extended Relational Database System for Engineering Data Management. IEEE Database Engineering, 7(2), June 1984.

[Weinreb 1981] Lisp Machine Manual. Symbolics, Inc., 1981.

[Wilkins 1984] M.W. Wilkins and G. Wiederhold. Relational and Entity-Relationship Model Databases and VLSI Design. IEEE Database Engineering, 7(2), June 1984.

[Wirth 1982] N. Wirth. Programming in Modula-2. Springer-Verlag, New York (NY), 1982.

[Woelk 1987] D. Woelk and W. Kim. Multimedia Information Management in an Object-Oriented Database Systm. In: 13th VLDB conference, Brighton, 1987.

[Zilles 1984] S.N. Zilles. Types, algebras, and modelling. In: M.L. Brodie et al., editors, On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages, Springer Verlag, New York (NY), 1984.

[Zloof 1977] M.M. Zloof. Query-by-Example: a database language. IBM Systems Journal 16(4), 1977.

BLANK PAGE

SPATIAL ANALYSIS

BLANK PAGE

Towards A Spatial Theory*

Andrew U. Frank
Surveying Engineering Program
University of Maine
Orono, ME 04469, USA
FRANK@MECAN1.bitnet

Abstract

In several recent publications, problems with spatial reasoning and the lack of a coherent spatial theory have been noted. Human beings are very good at analyzing spatial situations, determining patterns and similarities, etc.; however, programmed systems on computers cannot compete.

The well-known Euclidian geometry is not a true image of how human beings think about geometry. The ideal objects of Euclidian geometry are quite unlike the real, extended objects of which the world is made. Moreover, Euclidian geometry cannot be easily modelled on computer systems. The absence of a coherent spatial theory hinders development of GIS. Building a comprehensive, coherent spatial theory will not be simple. Several different approaches are currently tried: (1) defining algebras with finite representation that can model Euclidian geometry, (2) using combinatorial topology to represent geometry with simplices, and (3) applying models based on finite resolution rasters.

It is important that multiple approaches are tried and ultimatetly combined to best accommodate the concepts human beings use. In cognitive science, efforts to better understand spatial concepts will be contribute to a spatial theory.

1 Introduction

The development of computer systems to the present level of performance and reduced cost make many applications possible which were not conceivable a number of years ago. Computers can now be used to perform operations that in the past were carried out tediously by hand. We can also see new applications which were not

---

*This work was partially funded by a grant from NSF under No. IST-8609123

possible with the traditional technology. This development does not only influence the technical solutions we select or the problems we can successfully attack, but it will also influence the scientific concepts we use [Cebrian 1987]. This paper deals in the foreground with practical problems and possible technical solutions, but the methods proposed can be used to further the theoretical understanding of spatial situations and processes.

Spatial information systems, used as the common name for geographic information systems and land information systems [FIG 1981] are in a way a combination of traditional tasks and new concepts. Several decades ago, the possibility for this development became visible and a large number of optimistic predictions have been written; however, an assessment today reveals that only few system are productive and their use, although beneficial, is much less general than thought possible.

The development of software for spatial information systems falls behind user demands and some of the available programs still show major shortcomings. For example, none of the commercially available GIS software systems contain an algorithm for polygon overlay which is absolutely foolproof. The best ones work in practical cases, but some fail even with relativly simple, practical problems. A recent discussion with a development team confirmed that the programming of geometric algorithms is the major difficulty. It was generally believed that the geometric operations were simple to program—after all, "this is (2-dimensional) high school analytical geometry". Unfortunately, this is not correct and we begin to understand our errors. We stil do not completely understand the tasks that human beings and computers do well. Presentlt, human reasoning exceeds the computer's ability to analyse and deal with spatial situations. This paper attempts to outline a research program to understand the human capabilities better, and find possibilities to model them in computer systems.

Too often computer geometry and computer graphics have been lumped together. In this report we separate the concerns of processing geometric data from considerations for output presentation. Clearly, graphical presentation methods are based on geometric processing, but may have different requirements than processing of geographic data for analytical purposes.

As a second level of separation, we advocate the analysis of geometric aspects of information in a separate step from the treatment of the associated attribute data. Clearly a spatial information system must contain routines to treat both classes of data, but a separation of concerns will simplify the design and coding [Frank 1987]. This paper deals with the spatial aspects only.

2 The Concept Of Spatial Theory

Boyle [Boyle 1983] reports on a NASA sponsored conference about the "review and synthesis of problems and directions for large scale cartographic information system development." The conference report contains an extensive discussion of the need for a spatial theory—a term which was first used by Chrisman [Chrisman 1975]:

> "There is at present no coherent mathematical theory of spatial relations.
>
> The world in which we live is a spatial world and to exist in it all of us master spatial relations on an intuitive basis. While these relationships are highly complex, we usually need not even think about them. We make use of the eye/brain combination to sort out the visual images we receive and recognize their various elements.
>
> In dealing with spatial data in the form of maps, images and the like, we usually obtain the data with our eyes and then deal with spatial aspects of the data on an intuitive basis.
>
> Unfortunately, when a computer must be instructed in how to perform such operations, intuition must be replace by precise statements and by precise mathematical relationships.
>
> At present we cannot supply this need.
>
> The lack of a coherent theory of spatial relations hinders the use of automated geographic information systems at nearly every point. It is difficult to design efficient data bases. Difficult to phrase queries of such data bases in an effective way, difficult to interconnect the various subsystems in ways which enhance overall system function, difficult to design data processing algorithms which are effective and efficient. As we begin work with very large spatial data bases or global data bases the inabilities and inefficiencies which result from this lack of theory are likely to grow geometrically.
>
> While we can continue to make some improvement in the use of automated geographic information systems without such a coherent theory on which to base our progress, it will mean that the development will rest on an inevitably shaky base and progress is likely to be much slower than it might be if we had a theory to direct our steps. It may bee that some advances will simply be impossible in the absence of a guiding theory." [Boyle 1983]

A spatial theory must therefore include the following points:

- Define a set of generic geometric objects and generic operations on them. Special attention must be directed to include special cases and error conditions.

- The generic objects and operations must be suitable for modelling the geometric properties in a spatial information system; that is, the higher level objects are distinguished by their attributes which restrict and redefine the geometric operations applicable. The quality of the theory is judged by its capability to reduce redundancy and possible inconsistencies in the definiton of specific geometric objects.
- Explain the behavior of commonly used operations on geometric objects in terms of the previously precisely defined objects and operations. The success of these definitions can be measured if the commonly used operations can be described easily.
- A spatial theory should be as general as possible, thus ensuring a wide range of applications. In the following proposal, we seek for a theory which is independent of the dimensions of the objects it is applied to (i.e. it can be used for the two or the three dimensional case).

2.1 Need For A Spatial Theory

A coherent spatial theory should tie together the different concepts of space which are used by the builders and users of a spatial information system. The differences in concepts between the mathematically inclined, computational geometry research and the practical users of systems are wide, and the gap is difficult to bridge. The differences appear in the user interface where users are surprised by system reactions because the concepts they use and the spatial concepts embedded in the system differ substantially. Building comfortable interfaces is only possible if a clear understanding of the user concepts is available, or a set of coherent concepts, which are used for the design of the interface, can be taught to the user. In order to have the system perform, the operations on the user level—and such operations may be different for different classes of applications—such concepts must be general enough that they can be used internally for the design of the basic geometric software. The development of geometric software is so difficult and thus expensive, that a generalized solution can be developed which is useful for spatial information systems as well as for all physical CAD/CAM systems for mechanical and civil engineering.

Clear definitions for spatial concepts are also crucial for data exchange. Transfer of data from one system to another is difficult if the model used for spatial representation differs significantly. A standard for exchange of spatial data needs a formal definition of spatial concepts which can be used to establish conversion routines.

2.2 What Spatial Theories Are Available

A number of theories deal with space from different points of view. We often revert to the high school form of Euclidian geometry and very few know of "elementary geometry from an advanced standpoint." [Moise 1963] Mathematicians have further studied topology, including graph theory (which is often used in spatial information systems). Analytical geometry shows how to transform geometric objects and operations to the realm of computations (and computers).

Geodesy deals with some aspects of physical space and how they can be measured precisely, whereas GIS often uses rasterized methods to approximate the spatial distribution of some properties. Physics, in general, includes some aspects of the geometric properties of objects—the recent development of "naive physics" [Hobbs 1985] attempts to capture some of the common understanding of geometric properties of reality with less abstraction than is evident in the pure theory.

3 Classical Geometry vs. "Naive Geometry"

Geometry deals with abstract objects, like infinitely small points, lines which extend into infinity, etc. Physical objects in the real world are very different: points have a finite size, lines are finite and never truly straight or parallel. Moreover, our abilities to measure are limited and only approximations for the true values can be determined.

In this section, first the concept of geometry are discussed as they are seen by classical and modern mathematics before extending it to take into account the restrictions of real physical objects.

3.1 Classical View Of Geometry

Classical geometry, as developed by the Greeks and usually taught in high school, deals with abstract objects (primarily points and lines) and their interaction. All operations are restricted to be performed, at least in principle, by compass and straight edge.

These abstractions have been very helpful for a simple axiomatization of geometry, but it must be remembered, that there are no objects in reality like points and lines, or relations like parallel.

3.2 Concept Of Geometry

Geometry in modern mathematics is broader than Euclidian geometry. The aim is to determine the abstract concept of geometry. Since the work of Felix Klein [Klein 1872] it has been accepted that geometry deals with invariances of figures under transformations. This notion seems to capture more of what humans understand by "geometry" than the classical operations with points and lines. It is evident, that not only the figures 1a and 1b are related, but most will also see a relation between figures 2a and 2b. The figures 1 are related by similarity transformations (translations, rotations and scaling), whereas the figures 2 are related by more general topological transformation.

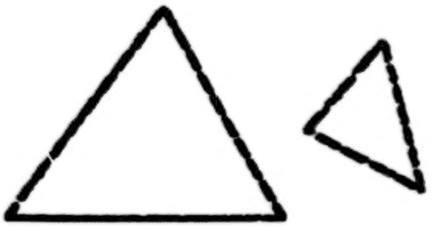


Figure 1 (a) and (b)

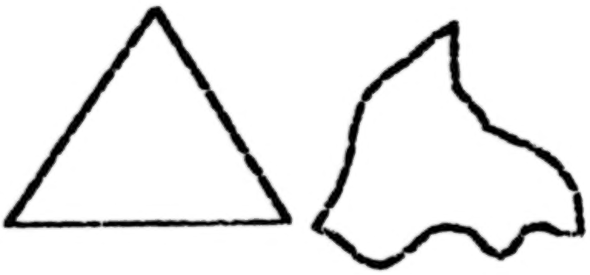


Figure 2 (a) and (b)

In a spatial information system we often want to search for similar patterns, but do not expect to find the same geometric figure again. A number of geographical studies, for example in migration analysis or geomorphology, try to find similar spatial patterns. It is an important task for a spatial theory to explain what "similar" means in each of these cases. There have been attempts to define patterns by using symbolic processing based on a triangular surface [Frank 1986]. The class of continuous transformations in mathematical geometry is too broad and limited. Sometimes, we judge two figures, which are linked by a continuous transformation, as quite different (figure 3a and 3b), or we judge two figures as similar, even if they are topologically different (figures 4a and 4b) [Walker 1987].

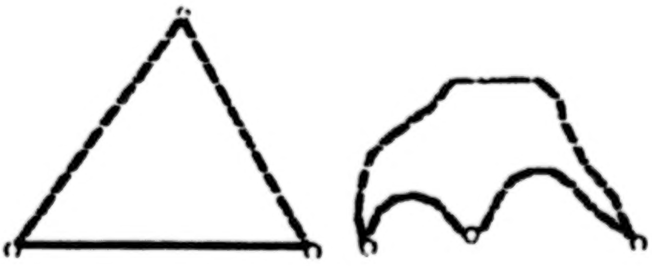


Figure 3 (a) and (b)

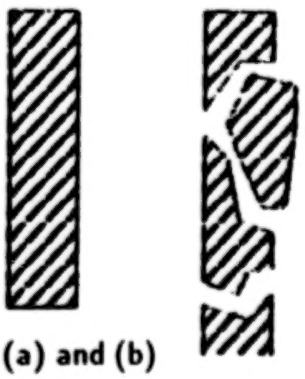


Figure 4 (a) and (b)

3.3 Naive Geometry

The objects we deal with daily are quite different from the abstract geometric concepts we encounter in the mathematical treatment of geometry. The extension of physical objects are limited, but most geometric concepts (e.g. line) are infinite. Secondly, geometry is based on the ability to measure without error, but all our measuring methods are inherently imprecise to a certain degree.

Our daily operations are not severely limited by these problems. As human beings we use a sort of 'naive' geometric reasoning, taking into account automatically the differences between abstract geometry and the geometric properties of real objects. Despite the fact that we all know how to do this, there is no formal understanding of this process. A formal description of this "naive" reasoning is necessary in order to have computers simulate such behavior. It seems worthwile to study if "fuzzy logic" [Zadeh 1981] applies to our reasoning about spatial relations ('close', 'next to', 'far', etc.).

In the arena of expert system construction, a similar disagreement was found between the laws of theoretical physics (i.e. what was learned in college in Physics I and II) and the concept humans apply in daily life—we take into account all the impurities of friction in motion without much trouble. The attempts to formalize this "naive physics", so as to be incorporated into an expert system has been difficult, but important in areas like building expert systems to assist in the diagnosis of faults in mechanical systems (e.g. railway engines).

A spatial theory should include a "naive geometry" in the above sense. It must be expected, that this would also be helpful for the development of the "naive physics", because much of physical reasoning is based on geometry.

4 Mathematical Geometry vs. Modelling Power Of Computer Systems

Modelling of geometry in computers is most always attempted based on analytical geometry. Analytical geometry provides a convenient base for translating geometric problems into numerical ones, thus making them more suitable for computer processing.

Unfortunately, the use of analytical geometry for computer modelling of geometric figures has some serious shortcomings. Analytical geometry needs a mapping to real numbers. The finite approximations in a computer system are not sufficient. The floating point numbers (REAL) of a FORTRAN program are much more akin to the integers: there is only a finite number of them and between some of the floating point numbers, there are no others.

Mathematically speaking, there is a difference between the field of real numbers and the integral domain (a ring). The Euclidian concepts of geometric space are intrinsically linked to the properties of spaces built over real numbers. It was shown, that analytical geometry using computer approximations is not in every case invariant under translation and rotation—the minimal requirements in Euclidian geometry [Franklin 1984].

5 Raster Based Systems vs. Vector Systems

Many useful spatial information systems are not based on a geometric description of objects with vectors, but use a regular subdivision of space in a raster and then record the interesting properties for each raster cell.

Raster systems are based on a intrinsically different concept of space then a vector system. In a vector system space is the total of the extension of all the objects, each with its own identity, special form, and properties, and the space is organized as a function of these objects. In a raster system, space is an abstract concept which is subdivided in abstract cells, for which we know some properties.

In a raster system, objects can be approximated by a set of raster cells, but a raster system can not deal with the abstract geometric objects like points and lines, and must approximate them with small areas. Typically, line oriented operations are not possible, but a number of other interesting operations (shrink & grow) are available. This is primarily studied in image processing literature. In [Serra 1982] an algebraic treatment of typical raster operations (transformations) is proposed.

6 Spatial Relationships

In general we are not only interested in spatial objects, but also in the spatial relations between them. Again, we lack a comprehensive, formally defined set of spatial relationships.

6.1 Distance and Direction

In Euclidian geometry (generally in a metric space) there is a distance defined between any two points with the ordinary properties:

- $d(a, a) = 0$
- $d(a, b) = d(b, a)$
- $d(a, b) + d(b, c) \geq d(a, c)$

In an Euclidian space (generally in any vector space) there is also a direction defined.

Both of these operations are defined between two points. It is interesting to see, that distance and direction are not defined in the same type of space and are thus not invariant under the same transformations; however, both obey linear transformations to be made compatible with rotation or scaling.

Human beings use distance daily between extended areal objects to discuss the direction between two areas (Canada is north of Maine? - it is also to the east and to the west). It seems possible to treat distance and direction between extended objects using intervals and then apply interval relations which are closely related to the inclusion relations between extended objects. An obvious first idea is to replace the two objects by their "center" (e.g. their center of gravity) and then determine distance and direction between these. This does not adequately represent human beings' concept in a large number of cases. Peuquet proposes a method based on heuristics to deal with distance and direction between extended objects, which avoids the most obvious shortcomings [Peuquet 1987]. We experiment with a system based on an algebra of intervals.

6.2 Boolean Operations

For objects in a topological space, an intersection operation is derived from set intersection. It is possible to determine if an object is spatially included in another, if they overlap or are disjoint. It is also possible to define multiple kinds of "touching" between objects; moreover, a complete formalization has not yet been achieved.

6.3 Natural Language Concepts For Spatial Relationships

Natural languages use a number of concepts to express the relationships between objects in space. Investigations are just beginning, which analyze the concepts and separate methods, specific to some languages, and others which seem universal [Mark 1987].

In natural language, human beings use concepts which are not precisely defined, like "far" and "near". Their interpretation clearly depends on the context of their usage (e.g. "near" applied to a gas station is different from "near" applied to a major airport), but are not intended to be precise even in a single context. It will be difficult to capture these effects and model them. Fuzzy logic [Zadeh 1981] was used to observe the usage of "near" and "far" in a set context [Robinson 1987].

We can assume that the concepts used to express spatial relationships in natural languages are the 'natural' concepts for Human beings. It is important to understand them, formally define them and use them in at least two contexts: as key words in query languages and in system where the human being receives guidance to find a location from an information system.

6.3.1 Query Languages

Useful GIS must provide an interactive facility to ask questions to the system and see immediately answers in the form of small maps. In this context, there must be a language to express the desired information, including spatial relationships between objects, in order to specify the desired objects in a query to a GIS (e.g. show all the camping areas north of Bangor). In order for such a facility to work, the human understanding of the spatial relationships used, and the formal definitions applied by the system, must be very similar, otherwise the system will not retrieve and display the desired information, This may lead to frustration of the user if detected, or to erroneous judgement by the user if not detected.

6.3.2 Navigation Aids

The use of GIS to help humans navigate in complex situations seems to be an obvious application. Experiments with systems to navigate in towns to find locations specified by street address are underway, and the first commercial systems are available. It is expected that most cars in the future will include some navigational aid.

In such a system the user must be informed about spatial actions ("turn left at next red light"). It is not yet clear what type of concept for expressions of spatial relationships is most effective in this situation.

7 Possible Approaches

In this section we rapidly enumerate a number of approaches that are currently tried without assessment.

7.1 Other Number Systems

The work currently done at the Renselear Polytechnic Institute by W. Randolph Franklin attempts to overcome the problem that real numbers used in finite computer systems cannot correctly represent geometric situations. Rounding of computed coordinate values cannot be avoided. They study other number systems, e.g. (big) rational numbers [Franklin 1987] or number systems including roots of polynomials, which could overcome these limitations [Franklin 1984].

7.2 Raster Systems

Operations in current raster systems are more based on opportunity than on mathematical analysis. A most extensive treatment is found in [Tomlin 1983], but a more compact notation of the algorithm could show similarities in the proposed operations. Approaches to build an algebra of raster operations are shown [Serra 1982].

7.3 Combinatorial Topology

The application of combinatorial or algebraic topology to GIS has a long tradition [Corbett 1979]. Combinatorial topology offers concepts to deal with the topological relations between spatial objects in a symbolic form. By using a restricted form of a simplicial complex, we found that qualitative aspects of most of the metric properties can be included [Frank 1986a].

8 Conclusions

Understanding spatial situations and processes is a primary goal of most studies in geography. We clearly need a formally defined vocabulary that allows us to discuss and reason on these processes. By spatial theory we understand a set of concepts of abstract space and abstract operations on space, which are formally defined; however, such concepts are only valuable if they correspond to human beings' understanding of space. It is thus necessary to combine the formal theoretical studies with results from cognitive scientists, who analyze the 'natural' concepts we use.

Acknowledgement

Thanks to Jeff Jackson and Max Egenhofer who helped prepare this paper.

References

[Boyle 1983] A.R. Boyle. Final Report of a Conference on the Review and Synthesis of Problems and Directions for Large Scale Geographic Information System Development. In: NASA Contract NAS2-11346, April 1983.

[Cebrian 1987] J.A. Cebrian de Miguel and J.B. Sendra. Microordenadores en Geografia. In: 1 conferencia Latino america sobre informatica en Geografia, San José, Costa Rica, October 1987.

[Chrisman 1975] N. Chrisman. Topological Data structures for Geographic Information Processing. In: AUTO-CARTO II, 1975.

[Corbett 1979] J.P. Corbett. Topological Principles of Cartography. Technical Report 48, Bureau of the Census, Department of Commerce, 1979.

[FIG 1981] Fédération Internationale des Géomètres. In: XVIe Congrès International des Géomètres, Montreux (Switzerland), 1981.

[Frank 1987] A. Frank. Overlay Processing in Spatial Informaion Systems. In: N.R. Chrisman, editor, Eighth International Symposium on Computer-Assisted Cartography, Baltimore (MD), 1987.

[Frank 1986] A. Frank et al. Formal Methods for the Accurate Defintion of some Fundamental Terms in Physical Geography. In: Second International Symposium on Spatial Data Handling, Seattle (WA), 1986.

[Frank 1986a] A. Frank and W. Kuhn. A Provable Correct Method for the Storage of Geometry. In: Second International Symposium on Spatial Data Handling, Seattle (WA), 1986.

[Franklin 1987] W.R. Franklin and P.Y.F. Wu. A Polygon Overlay System in Prolog. In: N.R. Chrisman, editor, Eighth International Symposium on Computer-Assisted Cartography, Baltimore (MD), 1987.

[Franklin 1984] W.R. Frank. Cartographic Errors Symptomatic of Underlying ALgebra Problems. In: International Symposium on Spatial Data Handling, Zurich (Switzerland), August 1984.

[Hobbs 1985] J. Hobbs and R.C. Moore. Formal Theories of the Commonsense World. Ablex, 1985.

[Klein 1872] F. Klein. Vergleichende Betrachtungen über neuere geometrische Forschungen (Comparing Considerations about Recent Geometric Researches). Erlangen, 1872.

[Mark 1987] D.M. Mark et al. Spatial Terms and Spatial Concepts: Geographic, Cognitive, and Linguistic Perspectives. In: International Geographic Information Systems Symposium: The Research Agenda, Crystal City (VA), November 1987.

[Moise 1963] E. Moise. Elementary Geometry from an Advanced Standpoint. Addison-Wesley Publishing Company, 1963.

[Peuquet 1987] D.J. Peuquet and Z. Ci-Xiang. An Algorithm to Determine the Directional Relationship Between Arbitrarily-shaped Polygons in the Plane. Pattern Recognition, 20(1), 1987.

[Robinson 1987] V.B. Robinson and R.N. Wong. Acquiring Approximate Representations of Some Spatial Relations. In: N.R. Chrisman, editor, Eighth International Symposium on Computer-Assisted Cartography, Baltimore (MD), 1987.

[Serra 1982] J. Serra. Image Analysis and Mathematical Morphology. Academic PRess Inc., 1982.

[Tomlin 1983] C.D. Tomlin. Digital Cartographic Modeling Techniques in Environmental Planning. PhD thesis, Yale University, 1983.

[Walker 1987] J. Walker. The Amateur Scientist: Reflections From A Water Surface Display Some Curious Properties. Scientific American, 256(1), January 1987.

[Zadeh 1981] A. Newell. Test-Score Semantics for Natural Languages and Meaning Representation via PRUF. International Journal Man-Machine Studies, 10, 1981.

BLANK PAGE

Interpolation and Uncertainty in GIS Modeling

Kenneth C. McGwire and John E. Estes

Remote Sensing Research Unit, Department of Geography
University of California, Santa Barbara

ABSTRACT

This paper discusses the incorporation of data taken from point samples with area covering data types found in a geographic information systems (GIS). The authors provide examples for two methods of interpolating point data to area covering surfaces and statistical methods for characterizing error for these surfaces. The moving averages technique of interpolation is found to be computationally efficient, however, establishment of optimal interpolation parameters is difficult and only a simple error statistic can be generated for GIS data layers created using this method. Kriging, a geostatistical technique, is a more demanding method for interpolation of surfaces and may be appropriate for certain environmental variables. Kriging is shown to provide valuable estimations of error throughout the surfaced data layer, reflecting spatial positioning of samples and variance observed in those samples. Techniques for the estimation of error in GIS data layers, such as those discussed in this paper, must be implemented in operational GIS's in order to provide information products and confidence measures that will form the basis for effective decision making with spatial data.

1. INTRODUCTION

This paper presents examples and analysis of the implementation of moving average and geostatistical interpolation techniques in a GIS environment. As the legitimacy of analysis using a GIS is inherently dependent on input data quality, the uncertainty associated with synthetically derived datasets must be characterized in order to determine the limits of data application. This paper limits itself to examining the use of two interpolation techniques for a GIS in the context of providing a product with known error characteristics. These techniques were chosen to represent methods for meeting two different levels of accuracy and error characterization requirements for interpolated products. The "jackknifing" method (Mosteller and Tuckey, 1977) is applied to assess the accuracy and stability of interpolated products generated by moving averages. Benefits of geostatistical methods in optimizing interpolation criteria are presented using an example from the implementation of the kriging technique in a GIS environment.

Geographic information systems provide an efficient medium for the storage, analysis, and presentation of diverse spatial data types. Representation of parameters in the natural environment may be available in the form of defined areas such as vector based polygons, regularly spaced measurements such as rasters, or as irregularly spaced point samples. The internally consistent spatial storage found in a GIS data structure lends itself well to environmental and cartographic modeling tasks using diverse data types. However, characterization of environmental variables and functions which overlay data types generally require continuous definition of attributes over an area. The area covering format of polygon and raster data meet this requirement, but incorporation of data from point samples with the area covering formats requires estimation for locations between sampling points. The use of techniques to interpolate values from point samples creates an inherent uncertainty which will be propagated through manipulations with these estimated values.

Interpolation of point measurements to an area covering format is based on the observation that, on average, points that are close together in space are more likely to have similar values of a property than points which are far apart (Burrough, 1986). When working directly from point samples, the analyst is given the responsibility of defining the spatial characteristics related to variance in the point data, often without the benefit of collecting the data himself.

A simple method for converting point data to an area covering format involves the extension of point measurements to polygons of constant values. Surrogate indicators might be used to determine polygon boundaries, as in the case of U.S. Department of Agriculture Soil Conservation Service (SCS) soil surveys where aerial photography is used to determine transitions between limited field samples. When surrogate indicators of transitions in value are not available, boundaries between polygons may be determined by minimum distances to known values as done when generating Thiessen polygons (Monmonier, 1982). However, area covering techniques which represent transitions in variables as homogenous areas separated by sharp boundaries are not suitable for representation or modeling of environmental parameters which change gradually. Thus, there can be a need to use smooth interpolation techniques for extending point measurements to cover the study area defined in a GIS.

The material which follows presents a discussion of interpolation within the context of the GIS environment. Use of the moving average technique of interpolation is discussed, and an example of estimating error for products derived from this method is presented. Kriging, a geostatistical technique is presented as an effective interpolation method for certain environmental variables which allows sophisticated characterization of error in derived products.

2. INTERPOLATION IN A GIS ENVIRONMENT

Many methods have been developed for the smooth interpolation of samples taken in two dimensions. Smooth two dimensional interpolation provides a GIS with the ability to characterize continuously changing environmental parameters. Smooth transitions in area covering techniques can be encoded effectively in a raster based data structure (Maffini, 1987). The most common example of smoothly interpolated data layers in raster geographic information systems is the digital elevation model (DEM). Vectorized representation of elevations in the form of contour lines are convenient for human interpretation, however, cartographic overlay techniques and environmental models implemented in a GIS generally require that all locations have defined values. Vector representation of surfaces using methods such as triangulated irregular networks (TIN) may provide a capacity similar to that available using a raster data structure, however, this paper only addresses the implementation of two specific interpolation methods in raster based systems.

The creation of a surface from point measurements using interpolation techniques introduces inherent uncertainty in the accuracy of the data layer. It is important that bias and error variance in GIS data layers be understood when assessing the value of information derived from analysis in the GIS environment. Walsh et al. (1987) document the introduction and propagation of inherent and operational errors for analysis of multiple variables in a GIS where combined error ranged from 71% to 83%. Thus, it is apparent that knowledge of error for GIS products is of great value in establishing confidence for conclusions based on GIS analysis capabilities.

The choice of a technique for surfacing point data will be driven by the quantity and quality of input data and the requirements for accuracy and rapid turn around time in generating the output product. In the case of a quick response mapping task, there may not be time to use techniques which require significant amounts of data acquisition and analysis. However, when producing estimates for high risk venture fields such as oil or mineral exploration, one cannot afford not to take this time. The two methods of interpolation presented in this paper represent two levels of requirements in interpolation objectives. The moving average technique is computationally efficient and applicable to many environmental parameters and sampling strategies, but this method does not allow for effective characterization of error. The kriging technique is useful in certain cases, but requires a significant amount of data and processing time. However, proper utilization of the kriging method allows for valuable representation of error in the interpolated product.

3. MOVING AVERAGES

The moving average technique of interpolation produces estimates at a new location by averaging existing measurements weighted by their proximity to the point in question. The samples to be averaged are typically limited to those which fall within a user defined maximum radius from the location to be estimated. The use of a search radius establishes the maximum range over which existing values will affect the point to be interpolated and limits the amount of computation to be expended on a given estimate. Some algorithms also enforce collection of a minimum or maximum number of samples for averaging or a minimum number of samples in a set of directions. The weighting function applied to averaged samples is generally inversely proportional to the distance from a sample to the point being estimated. In using the moving average technique, the search radius and weighting function are chosen based on knowledge of data characteristics and position of the acquired samples.

The primary benefit of the moving average method is the computational efficiency of this technique. Point measurements can rapidly be surfaced to cover a study area. Difficulties with the moving average method of interpolation include appropriate definition of interpolation parameters, excessive smoothing of data planes, and an inability to extrapolate beyond the range of available samples. Establishment of interpolation parameters for search radii and weighting functions is left to the analyst, resulting in choices which may or may not be optimal for a given dataset. Iterative attempts may be necessary in order choose an adequate combination of parameters. The use of a moving average technique smoothes data values, resulting in a reduced dynamic range for the interpolated product. Locations where sampling data exist will be effected by averaging with neighboring values. Some moving average algorithms ensure that defined sampling locations maintain original values, but these methods can produce unrealistic spikes and pits in the surfaced data plane in order to meet this constraint (Burrough, 1986). Finally, reasonable extrapolation of the surface beyond existing measurements is not possible with a simple moving average, as the averaging technique requires locally defined transitions in data.

In order to analyze the influence of varying search radii and weighting functions on the moving average technique for surface interpolation, a statistical technique called jackknifing (Mosteller and Tuckey, 1977) was implemented in a GIS and applied to interpolated datasets to characterize the bias, variance, and stability associated with differing interpolation parameters. Jackknifing refers to the method where a sample is iteratively predicted from the remaining set of data with one data point omitted. This statistical technique provides a method of estimating variance for values at unsampled locations when using a particular combination of interpolation parameters.

The jackknifing technique was implemented on a dataset contained in a GIS which was developed for the Naval Air Engineering Center (NAEC) at Lakehurst, New Jersey. Interpolation techniques were used on monitoring well data at the NAEC facility in order to establish water table elevations in an area where jet fuel spills may threaten local well water supplies and nearby cranberry agriculture. Determination of water table gradients provides insight into subsurface transport characteristics for contamination events. A personal computer based ERDAS GIS and image processing system provided storage of land use data, monitoring well locations and values, and interpolated ground water surfaces. The ERDAS system also provided raster processing functions and a moving average method for interpolation of point values into a surfaced raster data layer. Repeated execution of existing ERDAS interpolation routines using a batch processing function provided a jackknifed dataset of interpolated values from digitized ground water elevations at the NAEC facility.

Three sets of interpolation parameters were tested on jackknifed values for the Lakehurst monitoring well dataset. An initial combination of a 130 meter search radius and linear weighting function proportional to the distance to a known sample divided by the total search radius (figure 1) was used based on examination of sample spacing and the desire for a smooth transitions in the output surface. Jackknifing interpolated values derived from the dataset resulted in a mean difference from the actual value of 0.03 feet and a standard deviation of 0.81 feet, indicating a slight (negligible) positive bias in the data but a fairly large degree of variation in values relative to the total range of the dataset (approximately four feet).

An adjustment in the second jackknifing trial utilized a search radius of 80 meters and the same weighting function. This change assumes that the error variance in the previous case was due to a suppression of the dynamic range of the output surface due to averaging with too many samples at a distance from the point in question. The mean of differences in jackknifed and actual

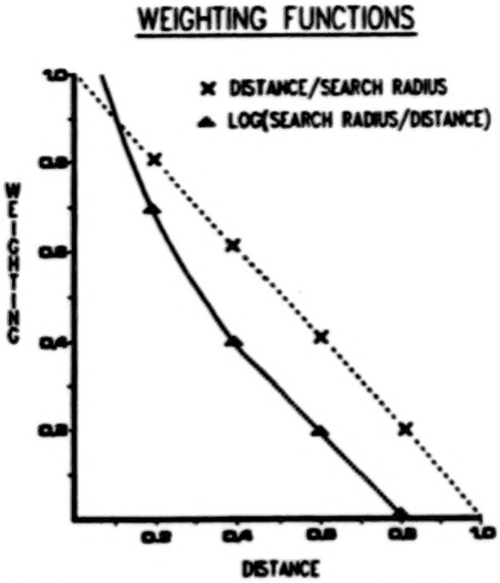


Figure 1: Weighting functions used on NAEC Lakehurst dataset.

values in the second case was 0.10 feet with a standard deviation of 0.83 feet. Rather than improving, we see that there is a slight increase in the bias and variance of estimations using these interpolation parameters.

Finally, a third attempt was made using a different weighting function with the 130 meter search radius. The new weighting function assigns weights proportional to the log of the search radius divided by the distance to the sample being used (see figure 1). These parameters resulted in a mean difference from the known value of 0.04 feet and a standard deviation of 0.78 feet. Variance in the interpolated product is seen to be the least using the third combination of parameters. Further iterations could be made with different values, but there is little guidance as to whether those values used in the third attempt indicate a shift towards an optimal set of interpolation parameters.

The values of variance provided by the jackknifing method indicate that the chosen interpolation parameters exhibit a significant degree of variance relative to the total range of measured values. These error estimates do not adequately represent the distribution of variance over the surfaced area. In the Lakehurst example, highly permeable, sandy soils can support a large volume of flow with a fairly small gradient in water table elevation. Therefore, the degree of uncertainty exhibited in these examples lowers confidence for estimations of subsurface transport based on spatial analysis of interpolated data.

BEST COPY AVAILABLE

While the jackknifing technique allows for analysis of the effects of search radii and weighting functions on the data to be interpolated, this method provides no guidance towards an optimal interpolation strategy. The implementation of geostatistical methods for interpolation establishes the ability to set optimal interpolation parameters for certain environmental variables based on positioning and variance in the dataset. In addition, the kriging technique provides valuable estimates of error for interpolated products.

4. KRIGING

Kriging, a technique of interpolation formalized by Matheron in 1970, was named after D. Krige, a South African geologist. This technique was developed for estimation of regionalized variables in mining applications. Regionalized variables are attributes whose value is related to position. The context of location in regionalized variables allows for effective application of statistical methods which are appropriate to estimations of environmental parameters. Kriging utilizes a spatial auto-correlation function developed from the dataset to produce both an estimate for a variable and a measure of error variance associated with that estimate.

Estes (1982) and others (Vitek et al., 1984; Walsh et al., 1987), have pointed out the need for accuracy statements to be attached to data in a GIS. The ability of the kriging method to estimate error associated with the spatial positioning of interpolated values is the prime benefit of using this method in the GIS environment. Knowledge of error variance associated with interpolated values allows determination of confidence limits, establishes limits on data value, and highlights locations needing further data collection. However, the kriging method is only useful for estimating those environmental variables whose variance shows a relationship with distance between samples. This condition is often met in the natural environment. Two major difficulties in using the kriging technique arise from and the need for a reasonably large dataset for estimating the relationship between data variance versus sampling distances and the computational demands in interpolating an entire raster grid.

A semi-variogram, the plotting of variance in pairs of observations against distance, is generated to describe random variation in a data set for use in the kriging technique (see figure 2). Assuming that the variance of measurements is a function of the distance over which observations are made, a model is developed to explain the empirical semi-variogram. This model of variance as a function of distance between observations provides guidance for the area to be sampled and the weighting function which should be applied to observations within this area when interpolating a new value. Estimation of the semi-variogram for a data set requires that underlying trends or "drift" in the data be characterized separately from the "random" component of variance which is explained by the semi-variogram. The technique of "universal kriging" allows drift in the data to be accounted for separately from the semi-variogram. Drift in the data is modeled using a selected polynomial function for the X,Y plane being represented. Calculation of drift is similar to techniques found in the global surfacing technique of trend surface analysis. The utilization of a mathematical polynomial to express drift is questionable in the case of complex environmental trends (Walden, 1972). However, for limited areas or environmental attributes which have gradual transitions underlying higher frequency variation, polynomial characterization of drift may be reasonable.

By utilizing the semi-variogram in determining interpolation parameters for search radius and weighting, kriging allows for the best linear unbiased estimate of a regionalized variable and produces minimum variance in interpolated values. The kriging technique also provides for "exact interpolation" in that the method guarantees that sampling locations will retain their original values (Royle et al., 1980). Because kriging provides the best linearly unbiased estimate of a variable, minimizes variance, and is an "exact interpolator", some authors refer to kriging as an optimal interpolation technique. However, Akima (1975) points out that this statement is misleading, as optimality is dependent on the application of the technique as well. As mentioned earlier in this paper, the choice of an appropriate technique is based on available data as well as the speed and accuracy required in output products. Though requiring greater sample sizes and analysis time than more simplistic methods of interpolation, kriging is quite valuable in situations where uncertainty is of great concern.

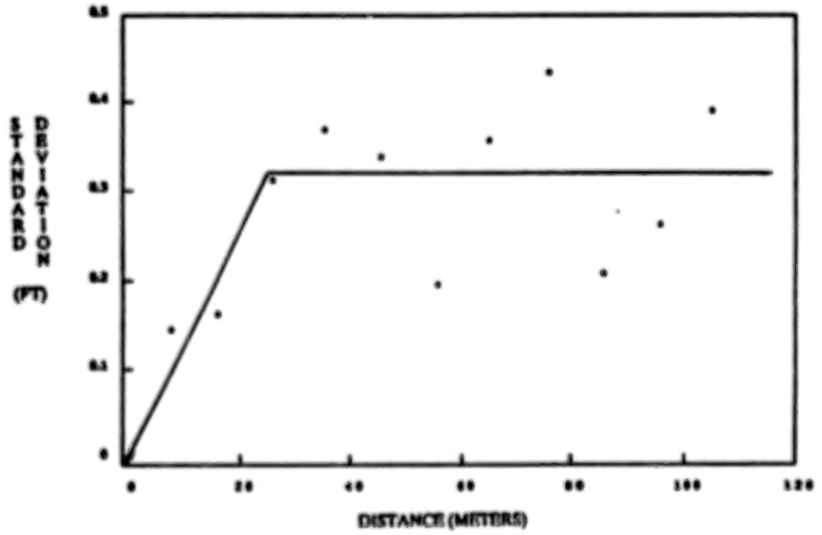


Figure 2: Semi-variogram for NAEC Lakehurst monitoring well data

BEST COPY AVAILABLE

The authors integrated a universal kriging program developed for the U.S. Geological Survey (Skrivan and Karlinger, 1980) with the ERDAS geographic information system. Input was modified to accept ERDAS format digitized points, and output was reformatted to produce geo-referenced data layers for both kriged values and their estimated variance. The kriging technique was applied to water table elevation data used in the NAEC Lakehurst study. Options in the kriging program which applied a technique similar to jack-knifing were used to establish the drift and semi-variogram model for the data. A first order polynomial was found to adequately characterize drift, and a linear model was developed based on the empirical semi-variogram. Figure 2 shows average variances for eleven segments of equal distance and a line representing the derived linear model of variance. In this case, an isotropic relationship between variance and distance was assumed, though anisotropic variance can be accounted for in the kriging technique.

Figure 3 displays land use in the Lakehurst study area. Of particular concern are the area immediately east of the hangars where fuel spills occur and the cranberry bogs at the southern edge. Figure 4 shows contours of ground water elevation derived from the kriged data layer. The error variance data layer associated with the interpolated surface (figure 5) shows that the interpolated ground water surface has small estimated standard deviations compared to the changes in ground water elevation. Thus it may be interpreted with a high degree of confidence that ground water transport of contaminants exhibits a consistent gradient towards the east/northeast, and the cranberry agriculture is not likely to be affected by contamination in this environmental pathway.

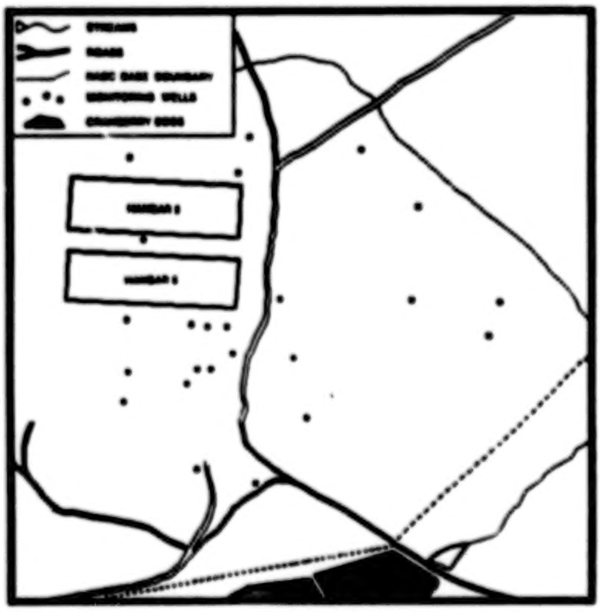


Figure 3: Land use for the NAEC Lakehurst study area

BEST COPY
AVAILABLE

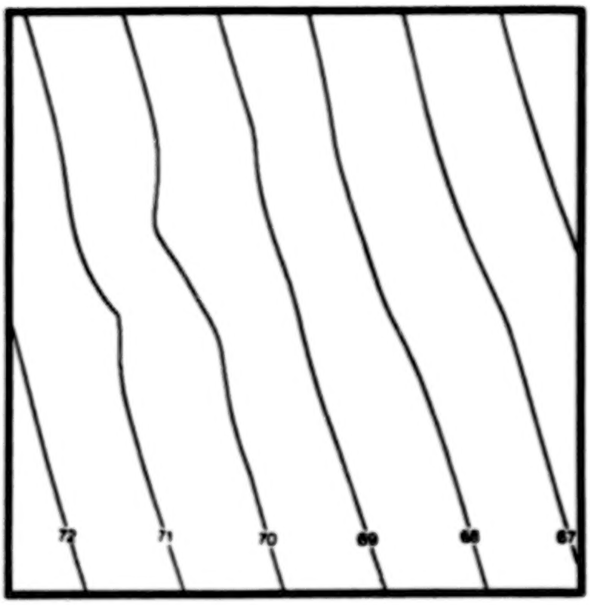

72
71
70
69
68
67

Figure 4. Kriged water table elevations at NAEC Lakehurst

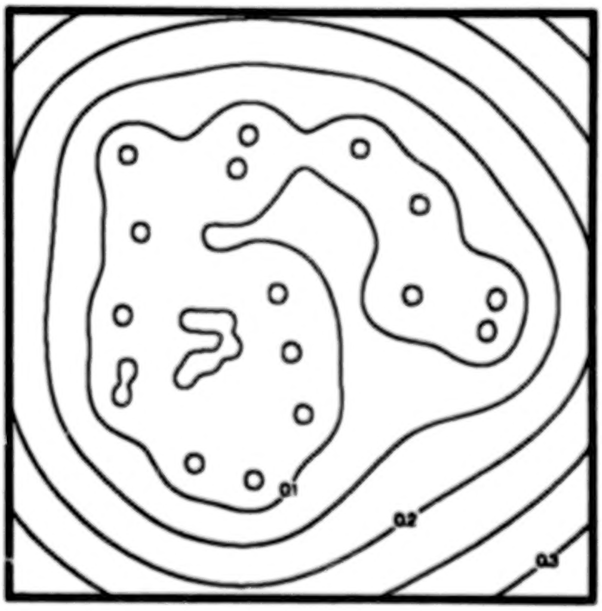


Figure 5: Standard deviations for interpolated estimates in figure 4

It can be seen in figure 5 that the estimated variance of the interpolated ground water surface is significantly lower than that expressed in the moving average examples. In addition the distribution of error that is related to the monitoring strategy is apparent. The cost of generating the kriged surface and its superior characterization of error is primarily in processing time. Surfacing the Lakehurst data set with the moving average technique took 33 minutes on an IBM PC/AT. Processing for the kriging method required 18 hours on the same machine. It should be noted that much time can be wasted by juggling parameters for the moving average technique.

Assuming sufficient observations exist, information from data layers in the GIS could be used to stratify data so that different moving averages or semivariograms could be applied to regions with different characteristics. Such a method would demonstrate the ability of a GIS to consider the spatial variability of factors affecting an environmental parameter when analyzing data for that variable.

5. CONCLUSION

A major question in any form of geographic analysis is how to use data from sample locations to generalize the spatial relationships present in a given area. The ability to go from "lumped" parameter models of point data to distributed parameter models is essential if we are to advance our understanding of geographic processes at all scales from local to global. The use of interpolation techniques in a GIS environment allows for the continuous definition of a value across a study area. Such a condition is required for incorporating point measurements with area covering data types when using overlay techniques or implementing environmental models within a GIS. Uncertainty that results from the extension of point measurements is of great concern in GIS analysis, and methods of characterizing error in these products must be developed. Simple interpolation techniques such as moving averages are easily applied to a dataset, but estimates of error for derived products involve a simple statistic for the interpolated data layer. Implementation of the kriging technique is possible for interpolation of certain environmental parameters. Though it is a more demanding method, kriging provides sound estimates of error which are linked to the spatial positioning in the interpolated layer. The implementation of an appropriate interpolation technique therefore depends on the quantity and quality of input data, the available data processing capability, and requirements for accuracy and development time for output products.

Several interesting applications can be found in the area of implementing kriging for geographic information systems. Information on the estimated variance in a data layer can be used to establish the degree of quantization which is reasonable for that layer, as the precision of measurement found at sampling locations should not be implied for the entire area. This can be seen in the error variance map produced in the kriging example (figure 5). The utilization of a semi-variogram may also be useful in the automated generalization of surfaces for data structures such as the quadtree (Samet, 1984) and TIN formats. Such a technique would allow spatial variance to dictate the degree of heterogeneity allowed in a parent mapping unit before division into sub-units would be statistically significant.

The need for reasonable assessment of inherent and operational errors in GIS based analysis is becoming increasingly evident. Effective decision making based on output from a GIS relies on confidence in the integrity of the information product. Further work must be done to develop a set of tools for dealing with uncertainty, and these tools must be implemented in commercial systems in order protect system users from false conclusions. The methods of jackknifing moving averages and kriging that are discussed in this paper show that a capability for assessing error in interpolated products exists, and that it is dependent on the chosen surfacing technique. These methods exhibit two levels of sophistication, and indicate the ability to utilize techniques which are appropriate to varying information requirements. Failure to develop techniques such as these may lead to the premature rejection of GIS technology as an effective decision making tool in the information systems marketplace.

REFERENCES

Akima, H. 1975. "Comments on 'Optimal Contour Mapping Using Universal Kriging ' by Ricardo A. Olea", *Journal of Geophysical Research*, vol. 80, no. 5, pp. 832-834.

Burrough, P. 1986. Principals of Geographic Information Systems for Land Resources Assessment; Clarendon Press, Oxford. 193 p.

Estes, J. 1982. "Remote Sensing and Geographic Information Systems Coming of Age in the Eighties"; Procs. of the Pecora XII Symposium, American Society of Photogrammetry, Falls Church, Virginia, pp. 23-40.

Maffini, G. 1987. "Raster vs. Vector Data Encoding and Handling, A Commentary", *Photogrammetric Engineering and Remote Sensing*, vol. 53, no. 10, pp. 1397-1398

Monmonier, M. 1982. Computer Assisted Cartography, Principals and Prospects; Prentice-Hall, New Jersey. 214 p.

Mosteller, F. and J. Tuckey. 1977. Data Analysis and Regression; Addison-Wesley, London. 588 p.

Royle, A., I. Clark, P. Brooker, H. Parker, A. Journel, J. Rendu, R. Sandefur, D. Grant., and P. Mousset-Jones. 1980. Geostatistics; McGraw-Hill, New York. 168 p.

Samet, H. 1984. "The Quadtree and Related Hierarchical Data Structures"; *ACM Computing Surveys*, vol. 6, no. 2.

Skriven, J. and M. Karlinger. 1980. "Semi-Variogram Estimation and Universal Kriging Program (K605)", U.S. Geological Survey, Water Resources Division, Tacoma, Washington. 98 p.

Vitek, J., S. Walsh, and M. Gregory. 1984. "Accuracy in Geographic Information Systems: An Assessment of Inherent and Operational Error"; Procs. of the Pecora IX Symposium, American Society of Photogrammetry, Falls Church, Virginia, pp. 296-302.

Walden, A. 1972. Quantitative Comparison of Automatic Contouring Algorithms; (technical report) Kansas Geological Survey, Geologic Research Section. 115 p.

Walsh, S., D. Lightfoot, and D. Butler. 1987. "Assessment of Inherent and Operational Errors in Geographic Information Systems"; Procs. of the 1987 ASPRS-ACSM Annual Convention, American Society of Photogrammetry, Falls Church, Virginia, vol. 5, pp. 24-35.

BLANK PAGE

ARTIFICIAL INTELLIGENCE

BLANK PAGE

ORGANIZATIONAL AND KNOWLEDGE BASE CONSIDERATIONS FOR THE DESIGN OF DISTRIBUTED GEOGRAPHIC INFORMATION SYSTEMS - LESSONS FROM SEMANTIC MODELING*

Vincent B. Robinson
(vbr%survey@uncaedu.bitnet)

Department of Surveying Engineering
The University of Calgary
Calgary, Alberta T2N 1N4
CANADA

C. Gustav Lundberg

School of Business Administration
Duquesne University
600 Forbes Avenue
Pittsburgh, PA 15282
USA

ABSTRACT

There is an increasing tendency for organizations to move from database to knowledgebase technology and an increase in distributed GIS. These two trends accentuate the relevance of semantic modeling in the successful application of GIS technology. We look at the relationship of organizational image to information system design and development. Semantics become important at the design stage of MIS. The relevance of general MIS experiences to GIS are discussed and linked to the importance of considering the semantics of spatial query and spatial statistics. It is suggested that intelligence and person-specificity of such systems should be concentrated in the query system, not in the database itself. This would suggest that much of the research into spatial query should continue, but within the context of knowledgebase systems .

1.0 INTRODUCTION

For some time now geographers have paid attention to the important distinction between geographic data and geographic knowledge. Early work in this area dates back to descriptive mental map studies that clearly illustrated the difference between objective and subjective environments. Later, a number of detailed studies built on this foundation, examining the intricacies of human distance estimation. More recently, some attention has

---

* This research is partially supported by the Alberta Forestry, Lands, and Wildlife Professorship in Digital Mapping and Spatial Data Management and a grant from the Natural Sciences and Engineering Research Council of Canada.

been paid to how humans represent and use spatial information. This research branch has resulted in a at least two emerging bodies of potential usefulness to GIS. One explores the relationship between natural representations and database query systems. The other explores naive spatial statistics and spatial problem-solving. In this paper we discuss how many of the findings in the area loosely defined as semantic modeling can contribute to the further development of GIS and where other may signal future complications.

There are number of trends which accentuate the relevance of semantic modeling in the successful application of GIS technology. One general trend is the evolution from database technology to knowledgebase technology (1). We couple this trend with the clear growth of distributed information systems which use GIS as the centerpiece of the system. This trend is particularly obvious among local government where 70-80 percent of their activiitites and information are location-related (2). GIS is becoming so important to some locationally sensitive organizations that to extrapolate the trend is to suggest that GIS is the kernel upon which tomorrow's MIS will be based.

2. DATABASE TO KNOWLEDGEBASE

Figure 1 depicts the differences and similarities between conventional database approaches and knowledgebases. We draw upon discussion in (1) to highlight the differences and similarities between database systems and knowledgebase (KB) systems. Database systems control the internal formats of permanently stored data. for the external formats on printers, displays, report writers, and screen managers they take the next step of separating output formats from the procedures. Figure 1 shows an application program connected to an I/O formatter and DBMS. Both of them are controlled by separate tables of descriptors. Although the external I/O and database calls are controlled by rules, they are not KB systems because they still use conventional programs for computation.

KB systems separate computational steps from the control flow and placing them in nonprocedural tables of rules. Figure 1 shows the inference engine as a small, fixed program that does not change when a new piece of knowledge is added to the system. Every fact is represented once only and every rule that needs it uses the same representation. The KB system outlined in Figure 1 has three separate rule-driven facilities that serve the user. Nameely they are (a) the language handler, (b) inference engine, and (c) DBMS that stores and retrieves data upon request. The table of conceptual schemata may be shared by both the language handler and the inference engine. Thus, the same knowledge that provides semantic basis for natural language can be used to guide deduction and problem solving.

At the conceptual level database systems are coming to be used like KB systems. That is to say to say they are being used to represent the corporate knowledge of an organization. One of the trends fueling this shift from database to knowledgebase is the increase in distributed information systems. To operate effectively, most distributed information systems require knowledge about the databases, their potential interactions, and how the access them. In other words, meta-knowledge regarding a number of database systems each of which may describe a particular subunit of an organization.

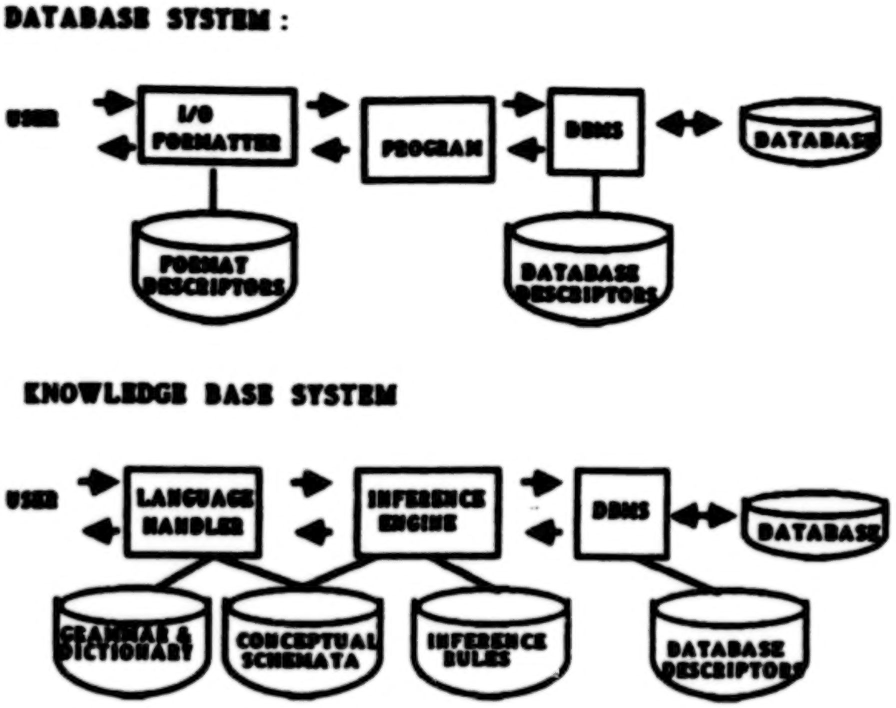

DATABASE SYSTEM :
USER
I/O FORMATTER
PROGRAM
DBMS
DATABASE
FORMAT DESCRIPTORS
DATABASE DESCRIPTORS
KNOWLEDGE BASE SYSTEM
USER
LANGUAGE HANDLER
INFERENCE ENGINE
DBMS
DATABASE
GRAMMAR & DICTIONARY
CONCEPTUAL SCHEMATA
INFERENCE RULES
DATABASE DESCRIPTORS

Figure 1. Contrasting Database Systems with Knowledgebase Systems (from (1)).

3. ORGANIZATIONAL IMAGE vs INFORMATION NEEDS

In this section we discuss conversions necessary to make information stored in a KB fit the requirements of specific, organizational domains. We also review two primary topics with an organizational bearing on distributed information systems. The dichotomy between language and activities, and the degree to which information systems mirror an organization's view of itself.

BEST COPY AVAILABLE

We feel that builders and users of GIS can benefit from work in progress under the umbrella of management information systems (MIS). In management information systems (MIS), model builders distinguish between the Enterprise Model (EM) and Integrated Data Base (IDB). As a corporate data model. The EM delineates a firm's business rules through a representation of relationships between logical entities. The EM defines the entire organization. Kerr (3) claims that an EM is a necessary requirement for the developemnt of an IDB which repesents the physical system. Drawing on experience from Pratt and Whitney, Kerr's design consists of two processes. The first process depicts (a) the business function being studied, (b) corresponding business functions, and (c) business views. Business views are defined as the packet of information passed between functions during the execution of a business procedure. On the other hand, a data model -

a. defines data items contained in each business view,
b. normalizes each view into logical entities,
c. determines entity relationships based on business rules,
d. graphically represents the model.

The bottom-up approach for building EM relies on a committment to a data-driven design methodology, a project data model focusing on inherent data relationships rather than procedural requirements, and development of a data dictionary system.

There are five steps to building an EM. First, it is necessary to review the current EM in order to pinpoint common data. The second step involves the extraction of common data from the EM by copying common data constructs from the model. It is important to include both entities with their corresponding data items and entity relationships since the collection of constructs becomes the touchstone of the the data model and the integration point for the EM. Third, project data model(s) is completed. Fourth, completed project data model(s) is integrated with the EM. This task involves updating the EM by adding the completed project data model to it. Finally, it is necessary to update the data directory. It is interesting that Kerr's (3) procedure exploits the difference in how project functions view the data to derive entity relationships between previously unrelated entities. In order to be accurate, the EM must represent the most inclusive view, even if the relationship is only recognized by one function.

Johnson (4) discusses the mediating role of Enterprise Analysis (EA) between business planning and MIS planning. The objective here is to achieve an optimal MIS portfolio by integrating business and MIS plans and then to select appropriate implementation technology. In this context there are four popular approaches for defining needs.

Business Systems Planning. In IBM's Business Planning System (BSP) a project team chaired by a user-manager analyzes information needs from the top-down to create a data plan for building information systems from the bottom-up. The system allows trade-off decisions that balance between short- and long-term projects, and starts its planning by defining the ideal system.

Critical Success Factors. The Critical Success Factors (CSF) approach focuses on interviewing top managers - discussing and clarifying goals and priorities, determining critical success factors, defining the prime measures of CSF's, report formats, and data recording (5).

Functional Stage Assessment (FSA). In FSA one identifies the organisation's current learning level and what the organization is capable of doing next (6).

Architectural Planning (AP). AP starts by defining existing support and then links corporate direction to MIS support via a planning matrix. Frequently, business organizations merge tools from different approaches in order to tailor its approach to allow for differences in culture, size, and organizational structure.

Domain differences place constraints on the applicability of these information needs/extraction frameworks for the development and use of GIS. BSP and CSF frameworks may be directly useful for a variety of GIS tasks. FSA and AP may be less directly applicable, then primarily in strongly organization-specific designs. Rather than directly adapting these techniques, we suggest that GIS designs at this development stage should be based on Protocol Analysis-based knowledge extraction procedures.

Interdependent meanings define an organization's concept of reality, a construct often labeled an organization's ontology. The challenge, therefore, is not managing facts, but rather managing meanings. A business rule (BR) is an explicit statement of a constraint within an organization's ontology. BRs do not exist in isolation, hence once a new rule is validated against existing rules, it becomes part of the ontology and all subsequent BRs must conform to it. The data in BRs are generally referred to as entities, their descriptors as attributes of entities, and the connection between entities are labeled relations. Appleton (7) describes a four stage procedure for discovering BRs -

Stage 1: identify a set of entities
Stage 2: identify a single attribute of each entity that allows for the identification of a unique instance of an entity
Stage 3: build an information model to validate and document the relations that exist between entities, and
Stage 4: identify additional attributes of interest about each entity.

Calkins and Marble (8) provide an example of how this procedure might work when designing a production cartographic database.

Many MIS reseachers view strict E-R models as the solution to the problem of semantic integrity in distributed systems. As long as the primitives are unambiguous this may well be true. A system can cope with many meanings associated with a fact, and intransitivities in relations. One can often simply add datum-meaning tuplets to the database. However, if the meaning(s) associated with a fact are ambiguous, e.g. the meaning of a swamp, we need a system designed to allow unstructured data. This area of research warrants more attention in both MIS and GIS domains. In addition, the GIS/MIS communities need to clarify the very close relationship between E-RM and well-known knowledge structures like frames and scripts. It may well be that the main difference between a ER model and a frame is the latter construct's emphasis on action. A frame seems to coincide quite closely with 'view' or 'subschema' - constructs that tailor a query to a user's particular needs. The query would pull out only the useful information from a large E-R structure.

4. GEOGRAPHICAL INFORMATION SYSTEMS IN ORGANIZATIONAL ENVIRONMENTS

While acknowledging there are several differences between the domain discussed by Kerr (3) and GIS, we feel there are some very useful issues to be raised. First, in the domain of GIS it may be unrealistic to anticipate a strong top-down attempt to database management and design. The uses, users, and system managers seem too diverse for a successful standardization from this direction. It may be slightly less utopistic to envision a system that can be enticed toward integration through use of simple and readily available DBMS. In a system that is driven from the bottom-up, with a small number of competing DBMS, a form of "system darwinism" may lead to efficient coordination. We think that a bottom-up system with a self-organizing bias best serves our current state of definition of GIS - GIS is what GIS-users do.

GIS are typically implemented in an organizational environment. The organization may be a government agency or a commerical firm. There is a trend toward developing distributed GIS, especially within local and state government. We have conceptualized a general framework for discussing distributed GIS within an organizational context.

Typically, each unit within an organization has developed and maintained a 'knowledgebase' that is to some degree peculiar to that unit's view of the world. This subunit knowledgebase contains the 'facts' used by the subunit to describe their view of the world and the 'rules' which they use in their reasoning process in arriving at actions to take. Therefore, it fits their needs, regardless of whether it is manual or automated. One of the common justifications for an organization to go through the entire process of establishing a GIS is to ostensibly reduce the amount of redundancy in spatially-oriented data that is maintained among the subunits of an organization. Generally the approach taken is to develop some central, common spatial framework on which to hang everything else. The implication is that the common spatial frame (CSF) is meaning-neutral. Rules for managing and manipulating CSF, combined with the 'facts' of CSF establish what we will call the common spatial knowledgebase (CSKB). That is to say it is supposed to be an objective, geometric framework that holds little or no meaning regarding real-world objects. The meaning of real-world objects is left the subunits, each of which might have a different definition in the CSF for the term. For example, in one state-level government it has been determined that there exist, at least, definitions of "parcel" which agree geometrically approximately 60-70 per cent of the time. Thus, one of the major tasks to map each organization's view onto the CSF. There is some effort in trying to 'integrate' the views so that inter-subunit connections can be made.

Combining the subunit's KB through transformation rules with appropriate CSKB procedures, one can say that a GIS has been formed. This KB based definition of GIS is perhaps premature, therefore we provide two very different definitions of Geographic Information Systems. The first is almost too pragmatic - GIS is what GIS-users do. It is surprisingly useful though from an organisational and knowledge acquisition viewpoint. The second definition, one that we will return to in our conclusion, and a much more ambitious one is that of GIS becoming formalizations of geographic knowledge and perception.

Despite the few successful applications of KB models to spatial problemsolving, and the tremendous potential of future work along those lines, the greatest value of KB models today corresponds to a definition 'GIS is what GIS-users do'. It is the first level of KE that is precise enough to indicate how the databases will be used, how they will not be used, by whom, hidden from whom, etc., but flexible enough not to put a technical straight-jacket on future applications. We suggest an exploratory round of user contacts where a developing understanding of the current and planned used of the database culminates in concise tasks suitable for Protocol Aanalysis (PA). PA has been used extensively in cognitive science research (9, 10) on tasks varying greatly in complexity. The knowledge acquisition framework is based on some very basic extraction and recording principles. It provides much room for improvisational analysis. Fortunately, sufficiently many analysis frameworks have been discussed in detail to give general design guidelines (cf. 11,12). Also, even novice knowledge engineers frequently are surprised by the fact that extraction procedures usually fall into place as s/he gains insight into the problem domain. What GIS-users do (or hope to do), analyzed in this manner will provide a good approximation of what GIS-users should demand of the GIS design.

5. LESSONS FROM SEMANTIC MODELING

Many of the lessons in GIS having to do with semantic modeling have been the result of work on problems of spatial query. In addition, cognitive science has begun to contribute a substantial body of research to the problem of modeling spatial semantics.

5.1 Spatial Query and Semantic Modeling

Egenhofer (13) suggests that an extended query language for treating spatial objects should include topological relations (e.g., disjoint, contains, inside, meets, overlap, intersect, common boundaries, and equal) and metric relations (distance and direction). The ambiguity and complexity of all the above concepts may be readily evident, but, save one exception, have not been rigorously investigated from a semantic modeling perspective.

BEST COPY
AVAILABLE

Robinson and associates have developed a Spatial Relations Acquisition Station for deriving fuzzy representations of concepts like 'near.' They have found that these natural language representations are quite person-specific, whereas their work on multiperson concept formation suggests that one can expect little agreement on the definition of quite simple spatial relations (14). It is important to note that these linguistic concepts are not intended to supplant existing knowledge-structures, but rather to be integrated into any one of them. Robinson's technique provides an opportunity to investigate semantic variation among and between GIS-users as well as the virtually unexplored topic of multiperson concept formation for use in a GIS environment. Much may be learned if the potential of natural language systems capable of representing approximate concepts can be exploited to study semantic error. Robinson and Wong's work is certainly suggestive of the kind of insight that can be gained by just such an approach (14).

5.2 Semantics and Spatial Statistics

Lundberg's experiments with structure perception suggest that we need to pay attention to semnatic biases that seem to exist in spatial statistics. He found that subjects attempting to evaluate dispersion in a pattern, objectively measured using a nearest neighbor index, tended to underestimate both clustered and dispersed patterns when evaluating patterns that we described as lacking semantic meaning (i.e. abstract point pattern) or with a neutral meaning (e.g. location of 20 major cities in a state). For data with announced semantic content (e.g. sites of violent crimes in an urban area), subjects tended to overestimate the value. It is suggested that patterns deviate from people's expectations of what is "true" about society in general are most likely to show the greatest estimate distortions. These are important results since subjects or users are sometimes asked to estimate certainty factors or probability statements. Thus, Lundberg has shown how unreliable these estimates can be in a spatial semantic domain.

5.3 Cognitive Science and Semantic Modeling

Techniques based on approximate reasoning may constitute a method for explicitly incorporating uncertainty in the management and retrieval of information in spatial information systems. However, the problem is not simply that of allowing degrees of truth, but of understanding how we integrate the interplay of many vaguely defined parameters and resolve it in some definite utterance (15).

A level of geometric conceptualization mediates between our naive representation of the physical world and the application of locative terms, whereas contextual factors like relevance, salience, tolerance, and typicality bear on the choice and interpretation of a locative expression. Frequently, context-related pragmatic 'near principles' explain characteristics of the situation of use. Transformations applying to the 'near' meaning are of two kinds -

a. shifts to another relation or 'sense shifts', and
b. gradual shifts such that the ideal meaning is 'almost true', i.e., a tolerance shift.

Herskovits (15) concludes that the simple-relations model is inadequate and outlines a framework that includes elementary geometric description functions that map a geometric construct onto one of its parts, some idealization of it, such as a point, line, plane, or strip, some associated "good form" obtained by filling out irregularities, some associated volume that the construct partially bounds, an axis, or frame of reference, or a projection.

Many of the breakthroughs in cognitive science are associated with successful utilization of top-down processes. Bottom-up processing is based on features of the input, while top-down processing is based on previous knowledge or experience and is illustrated by goal-driven reasoning. In top-down processing the outcome of lower steps is influenced by the outcome of higher steps. Ghatala et al (16) have explicitly investigated the effect of semantic and nonsemantic factors on the integration of verbal units in recognition memory, whereas the effect of meaningful interpretation of pictures and 2-D data displays has been studied by Rafnel and Klatsky (17) and Lundberg (18,19).

6. CONCLUDING REMARKS

In this essay we have discussed problems related to the maintenance of specific domains with special emphasis on modeling the semantic content. We are aware that we have raised many questions, but provided few tangible solutions. Many of our comments reduce to one set of interrelated conclusions:

We do not seem to know enough about

> The representation of the most basic spatial concepts to start attempts to store information in or near their natural language form.
>
> How personal perceptions should be built into distributed geographic databases.
>
> Goal-driven database use and query.

We suggest that cognitive science provides a set of useful and powerful tools for knowledge extraction, and that MIS has addressed the concepts of determining information needs and organizational information system planning and design.

We agree with many MIS researchers who stress that the intelligence and person specificity should be concentrated in the query system, not in the database itself. This would suggest that much of research into spatial

BEST COPY AVAILABLE

query should continue, but within the context of knowledgebase systems .

We suggest that researchers in GIS should pay more attention to concept clarification with respect to the interrelated notions of E-R models, frames, scripts, and schemata.

We also think that some promising frameworks have been developed within MIS for top-down and/or bottom-up integration of different information systems. We suggest that a bottom-up, or default-driven, framework may best fit a fragmented environment, like that of a geographically distributed GIS.

Given the wide variety of users and purposes, GIS may not be based on stored geographical representations of objects but must store an internal, abstract model of objects, from which different graphical representations can be derived upon demand (20). His remark continues to reflect two of our concerns -

> Given our sketchy knowledge of spatial semantics, should we include meaning-laden items in a distributed geographic database ?
>
> Given multiperson/multipurpose databases, should not the intelligence and semantics be incorporated in the query and communication system ?

References

1. Sowa, J.F. 1984. Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, Mass.

2. Somers, R. 1987. "Geographic Information Systems in Local Government: A Commentary," Photogrammetric Engineering and Remote Sensing, 53(10): 1379-1382.

3. Kerr, J.M. 1987. "Corporate Data Models Flourish from the Bottom Up," Computerworld, May 11, 1987, pp. 71-78.

4. Johnson, J.R. 1984. "Enterprise Analysis," Datamation, pp. 97-103.

5. Rockart, J.F. 1979. "Chief Executives Define Their Own Data Needs," Harvard Business Review, 57(2): 115-126.

6. Nolan, R.L. 1979. "Managing Crisis in Data Processing," Harvard Business Review, 57(2): 81-93.

7. Appleton, D.S. 1984. "Business Rules: The Missing Link," Datamation, pp. 145-150.

BEST COPY
AVAILABLE

8. Calkins, H.W. and D. Marble. 1987. "The Transition to Automated Production Cartography: Design of the Master Cartographic Database," *The American Cartographer*, 14(2): 105-119.

9. Ericsson, K.A. and H.A. Simon. 1984. *Protocol Analysis, Verbal Reports as Data*, The MIT Press, Cambridge, MA.

10. Lundberg, C.G. 1984. "Protocol Analysis and Spatial Behavior," *Geografiska Annaler*, 66B(2): 91-97.

11. Lundberg, C.G. and A.C. Barna. 1987. "A Knowledge Based Model of Commodity Trading Expertise," *International Jrnl of Modeling and Simulation*, forthcoming.

12. Voss, J.F. and T.R. Greene, T.A. Post, and B.C. Penner. 1983. "Problem-solving Skill in the Social Sciences," in Bower, G.H. (ed.), *The Psychology of Learning and Motivation - Advances in Research and Theory*, pp. 165-213, Academic Press, New York.

13. Egenhofer M. 1987. "Towards an Extended SQL for Treating Spatial Objects." Paper presented at Second International Symposium on Trends and Concerns of the Spatial Sciences, University of New Brunswick, July, 1987, Fredricton, NB.

14. Robinson, V.B. and R.N. Wong. 1987. "Acquiring Approximate Representations of Some Spatial Relations," *Proceedings* Eighth Intern'l Symposium on Automated Cartography, pp. 604-622.

15. Herskovits, A. 1985. "Semantics and Pragmatics of Locative Expressions," *Cognitive Science*, 9: 341-378.

16. Ghatala, E.S., J.A. Bell, J.R. Levin, and D.L. Truman. 1978. "The Effect of Semantic and Nonsemantic Factors on the Integration of Verbal Units in Recognition Memory," *Jrnl of Experimental Psychology: Human Learning and Memory*, 4(6): 647-655.

17. Rafnel, K.J. and R.L. Klatzky. 1978. "Meaningful-Interpretation Effects on Codes of Nonsense Pictures," *Jrnl of Experimental Psychology: Human Learning and Memory*, 4(6): 631-646.

18. Lundberg, C.G. 1985. "The Influence of Semantic Knowledge on the Perception of Spatial Structure," Paper presented at Annual Meetings of AAG, April, Detroit, MI.

19. Lundberg, C.G. 1986. "Perception of Spatial Structure," Paper presented at Annual Meetings of AAG, May, Minneapolis, MN.

20. Frank, A.U. 1982. "MAPQUERY: Database Query Language for Retrieval of Geometric Data and Their Graphical Representation," *Computer Graphics*, 16(3): 199-204.

BLANK PAGE

Artificial Intelligence Tools for GIS*

Andrew U. Frank
Douglas L. Hudson
Surveying Engineering Program
University of Maine, Orono, ME 04469, USA
FRANK@MECAN1.bitnet
HUDSON@MECAN1.bitnet

Vincent B. Robinson
Department of Surveying Engineering
University of Calgary, Calgary, Alberta T2N 1N4
VBR%SURVEY@UNCAEDU.bitnet

Abstract

The essential topics in Geographic Information Systems (GIS), spatial and temporal reasoning, are difficult and require powerful tools and a formal systems approach. Future GIS will include tools and methods developed in artificial intelligence research and will resemble today's expert systems. An expert system is defined here to be a computer-based mechanism that produces results comparable to those obtainable from a human expert in some field. Moreover, intelligent user interfaces and reasoning about data quality can also benefit from the power of expert systems.

1 Introduction

Future computer-based Geographic Information Systems (GIS) will employ methods and tools developed in artificial intelligence research and they will resemble what is now called an expert system. This will allow a GIS to provide great power and flexibility to a diverse user community.

This paper begins by discussing some of these methods and tools and their applicability for use in an 'intelligent' system. The assertion is made that the use of these methods and tools is indispensable for building an effective GIS.

---

*This work was partially funded by grants from NSF under No. IST-8609123 (principal investigator A. Frank) and No. IST-8412406 (principal investigator V. Robinson).

An intelligent systems paradigm is first introduced and is followed by some discussions concerning knowledge representation, search strategies and machine learning. Significant emphasis is placed on the use of expert systems which may assist the user in specific task areas such as geography, cartography, data quality, and spatial relations in general. The envisioned services of a GIS are so powerful that users will need some help in using them. The GIS itself is expected to provide that assistance as well.

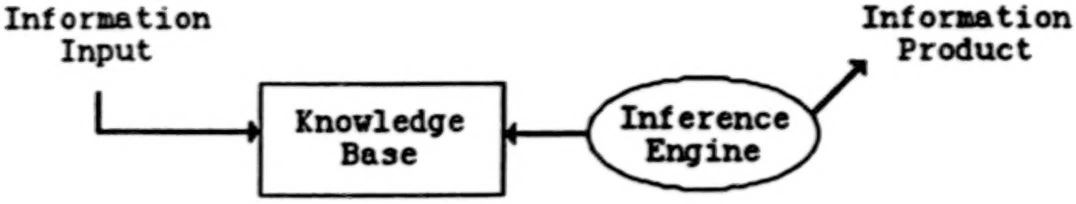
Some of the techniques and methods discussed in this paper apply generally to any information system, but they are most necessary for spatial information systems given the additional complexity of geometrical data processing and graphical output. In this paper, therefore, we will use the terms *Spatial Information System* and *Geographical Information System* without intending to differentiate between them.

2 AI Tools and Techniques

Artificial Intelligence (AI) research can be segmented into specialty areas concentrating in such subject matters as natural language processing, intelligent data retrieval, expert consulting systems, theorem proving, robotics, automatic programming, combinatorial and scheduling problems, and perception [Nilsson 1980]. Most of these are also areas of concern in the development of a GIS. Therefore, the tools produced to assist in the solution of AI problems are often directly applicable to GIS implementations, e.g. efficient search strategies, heuristic control of processes, interesting and efficient schemes for knowledge representation, etc. Some of these tools will be discussed in this section.

2.1 Intelligent Systems Paradigm

One of the most productive ideas to come from studies in AI is that of maintaining a definite separation between knowledge and the mechanisms which operate upon and control the processing of knowledge. This separation is not usually found in programs written in procedural computer languages such as Pascal or Fortran; in these, the results are obtained by describing the processes by which the results may be obtained. In an AI system results are obtained by simply stating what knowledge is required, leaving it up to the control mechanism of the system to infer the results in the best way it can. For this reason these control mechanisms are often called *inference engines*.



Separate Knowledge from its Control

Strict independence is not always easy (or sometimes even possible) to maintain. The advantage of this architecture, however, is that more powerful tools for operation or control can be developed independent of any specific knowledge domain. Moreover, new or modified knowledge domains can be constructed and applied to an existing inference engine to solve a variety of problems. Changes in the knowledge base (which are to be expected in any system which attempts to model reality) do not require corresponding changes to the general control machinery. More importantly from the user's view, it is possible to update the knowledge base without having to reprogram or even fully understand the internal operations of the AI system. Ideally the user only needs to understand a simpler model of the behaviour of the system, because inference engines are generally based upon a simple mathematical model describing very fundamental relations. It is usually not necessary for a user to comprehend the complexity of technical operations or to know the tricks-of-the-trade which are incorporated into the inference engine to enhance its performance. These latter concerns should be invisible to the user, freeing him or her to express relevant knowledge in a purer form.

Functional independence has obvious benefits to a GIS implementation. Geographic information is rarely static and, therefore, must undergo constant revision to remain current and accurate. This applies to factual aspects, e.g. that San Jose is currently the capital of Costa Rica, as well as to the rules of geography, e.g. that continental climates in temperate zones are characterized by hot summers and cold winters. Personnel more familiar with geography than with computer programming may modify the information component with relative ease if it is detached from programming concerns and if it can be formulated in familiar terminology. Complex programming problems which apply only to the inference engine can be addressed by people with the proper background in computer science. This partitioning allows each specialist to remain more productively in his or her own realm of expertise.

2.2 Knowledge Representation

Divorcing the knowledge base from the inferencing mechanism means that knowledge representation schemes are generally declarative in nature; it is only necessary to state what something is, not how to make one. Any procedural decisions are performed by the inference engine. The knowledge base is a collection of specific and general facts and descriptions of relations between them (e.g. Augusta is the capital of Maine; the area of Lot 6 is 2.45 acres; if the population of X is between 1000 and 25000 then call it a town; etc.). Most geographic information is of this kind. The format used for the conceptual representation of knowledge is very important as it determines what inference stategies can be used.

2.2.1 Logic

There is great interest in the use of logic for knowledge representation purposes. The first order predicate calculus (which is primarily declarative) easily lends itself to knowledge-based systems and there are readily available programming languages, such as Prolog and Lobster [Frank 1984], designed to work directly with the predicate calculus.

There are on-going discussions in the AI community concerning the adequacy and suitability of a first order logic for general knowledge representation [Winograd 1975] [Kowalski 1979] [Minsky 1981]. Many of the early arguments against it were guided by the perception that the predicate calculus is cumbersome and computationally inefficient [Newell 1981], but many of these problems can be overcome as more is learned about the controls that can be placed on the deductive process [Moore 1982]. Many other problems, however, are not yet satisfactorily resolved, for example those which concern time, uncertain or unknown values.

To resolve some of these questions multi-valued logics, those with more than just true or false as values, are being examined. Some interesting possibilities are: 3-valued logics, for instance using values TRUE, FALSE, and MAYBE, or TRUE, FALSE and UNKNOWN; probability logic in which a valid response may be that something has a .345 chance of being true; and Fuzzy Logic where values are determined according to interpretations of classification schemes.

2.2.2 Meta-level Knowledge

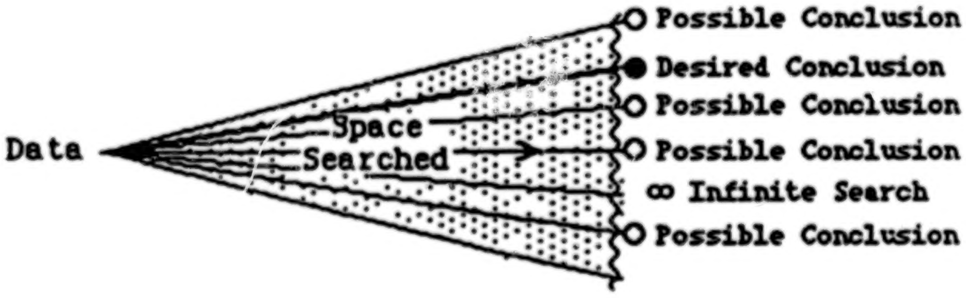
Meta-knowledge is knowledge about the organization of, or relationships among, other pieces of knowledge. There are meta-levels on top of meta-levels as well; no limits should be placed upon the levels of construction for meta-knowledge. Computational control can be based upon meta-level knowledge and it is imperative that meta-knowledge play an important role in the operation of an AI system. It is the only way in which the separation of knowledge base and inference engine can be maintained in non-trivial situations and it provides operational efficiencies the lack of which would make even small systems computationally intractable.

To incorporate meta-level knowledge it is important that the system provide support for basic knowledge abstraction. Procedures for classification, generalization, and aggregation need to be part of the system's design. Classification is the recognition that there is representative structure for individual objects, actions or concepts that share common properties; generalization is the meta-level extension of classification; aggregation is the recognition of the collective as an entity in its own right. Without these abstraction concepts the meta-knowledge will tend to be haphazard and difficult to comprehend and to use correctly.

2.3 Search Strategies and Heuristic Control

The database of information for a GIS is expected to be large and the mechanisms for accessing that information need to be powerful and efficient. Many of the early successes in AI research were developments of search techniques that make more efficient data retrieval possible and that can guide a search to optimal or near-optimal results. In general terms, a solution in an information system is found by searching through the database until all the relevant information has been found. Obviously the strategy of sequential search is only usable as long as the database is relatively small. For realistically large databases, more appropriate search stategies are required.

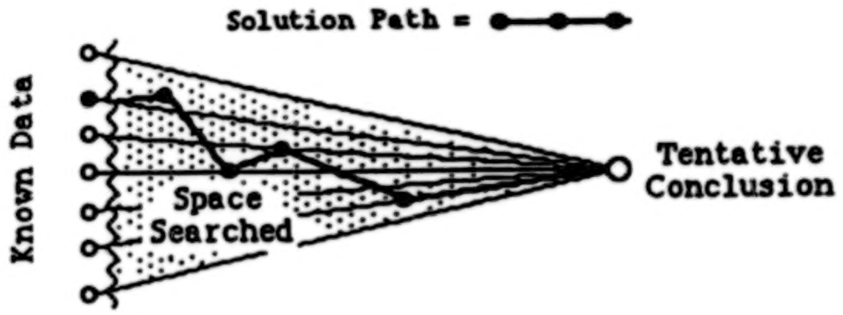
If all the possible solutions and intermediate steps to solutions of a problem are considered as states of that problem, the states can be visualized as a state-space or search space. By recognizing certain structuring possibilities in the state-space various search stategies can be advantageously employed. Often, if the state-space can be arranged hierarchically or as a graph, one of two basic search techniques, either forward or backward chaining, is used.



Forward chaining search space

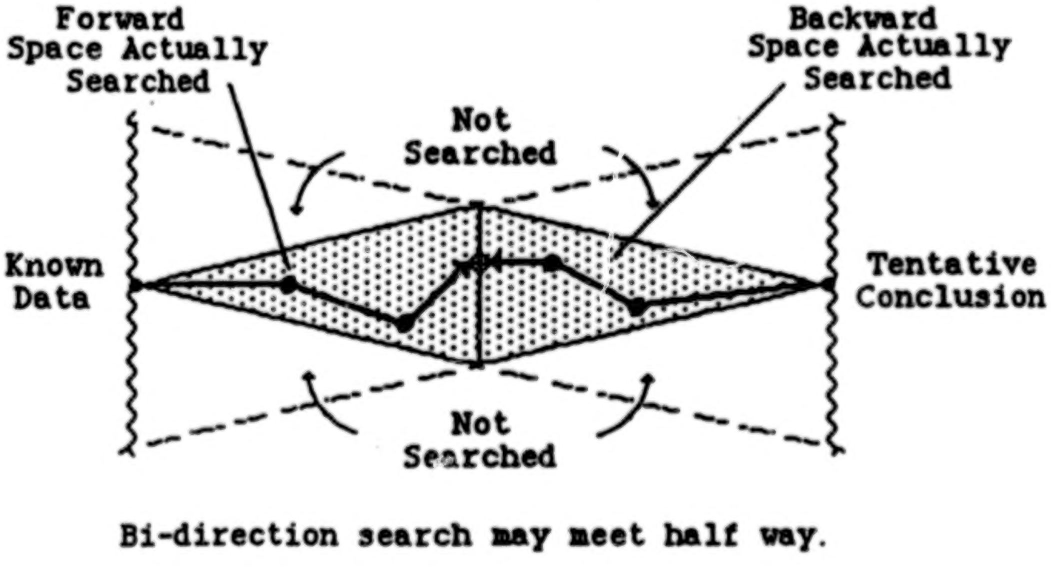
The former technique starts with some facts and rules and then searches for interesting conclusions which can be derived from them. It is sometimes called a data-driven search. If there are many branches in the search space, this technique may investigate many unproductive trails before some success is met. If the space is infinitely large, the search algorithm is undecidable and may never meet with success.

With backward chaining, a conclusion is tentatively proposed and an attempt is made to show that the conclusion can be derived from facts and rules stored in the database. The search space in backward chaining is usually, but not necessarily, finite and is considered to be more easily directed because the search is made from a known goal to known data values.


Solution Path =
Known Data
Space Searched
Tentative Conclusion

Backward chaining search space.

It may be possible to combine the two search methods to gain considerable efficiency. This is called a bi-directional search. The intersection of the two search spaces can be significantly smaller than with either technique alone.


Forward Space Actually Searched
Backward Space Actually Searched
Not Searched
Known Data
Tentative Conclusion
Not Searched

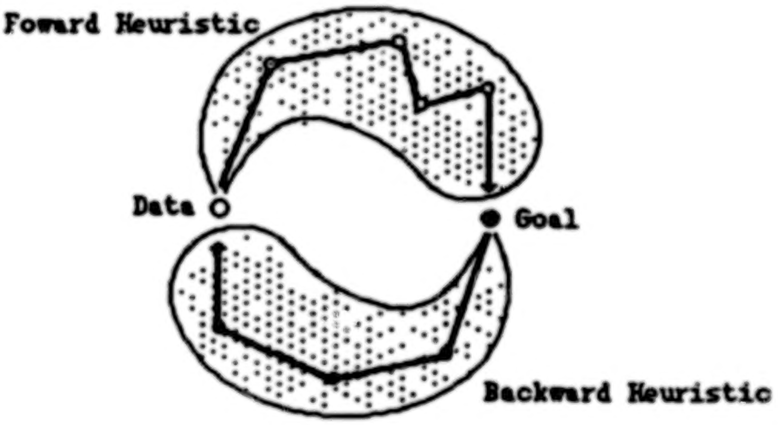
Bi-direction search may meet half way.

2.3.1 Heuristic Searches

In many instances the search directions can be controlled or assisted by additional information concerning the problem domain. Branch and bound algorithms, such as A* and AO* [Nilsson 1980], can often find optimal solutions without a full search of all those possible. Ocassionally heuristic methods are used to find solutions which may not be optimal. They may still be satisfactory, however, and can be obtained at costs far below more complete search methods.

A simple example of heuristic search control would be to cancel a request to search a long list of active military personnel for those who have reached the age of 150 years because there should be no values returned in this case. More complex examples can be found in chess playing programs where, from the many hundreds of moves that are possible, a particular one is selected based upon informed interpretation of current board positions and strengths.

Any information which will help to limit unproductive search, whether it is highly specific to the problem, a general rule-of-thumb, or just plain common sense, can be used as a heuristic. This technique has its disadvantages because there is often the chance that the best advice available turns out to be bad advice. For instance, in the case of a bi-directional search, heuristic control may cause the search paths to by-pass one another resulting in twice the work to find a solution.



Bi-directional heuristic searches may take twice the work of a single direction heuristic search.

2.3.2 Machine Learning

One of the most interesting areas of AI research concerns machine learning. It is a continuing theme that computer programs contain only the intelligence that the programmer and the knowledge engineer program into them; however, there are a number of examples of programs that have the ability to learn on their own. Example problems or constructions of increasing complexity are presented to these programs. The programs search for common properties and relationships among data entities, possibly comparing the new information with knowledge previous acquired. The new information is then integrated with the old, increasing the knowledge base.

Most of the questions in this area are still unanswered; however, there are a few initial efforts in this area concerning GIS. One of these is SRAS [Robinson 1987], a system concerned primarily with the acquisition of *fuzzy* natural language concepts of spatial relations. The KBGIS programs [Smith 1984] [Smith 1986] [Glick 1985] claim to have learning capabilities as well. Since the knowledge acquistion techniques currently used to create intelligent systems require the employment of highly trained (and expensive) human knowledge engineers, machine learning is a field that has enormous potential for growth as the recognition of the utility of expert systems increases.

3 Expert Systems

The most useful product of AI research is, and will for a long time continue to be, the concept of the expert system: a computer-based mechanism that produces results comparable to those obtainable from a human competent in a particular field of interest. Although a strict definition of an expert system is elusive, one should try to include in any attempt a sense of flexibility in the use of the knowledge under its control.

As with human experts, expert systems can be found with different levels of expertise. A modest system such as one which can determine simple relationships family relationships can be built in an hour or so and can be an important pedigogical tool for aspiring knowledge engineers. More sophisticated systems attempt such tasks as medical diagnoses, spectral analyses, and combat situation evaluations; these can take many people-years to construct.

Expert systems incorporate many of the tools introduced above to provide an appropriate response in situations where human experts are too costly or are not readily available. The expert system can provide advice in the acquisition and integration of primary data, assistance in optimal data storage, guidance in the appropriate operations to perform upon data, and direction in the ways and means of retrieving either primary data or conclusions that can be drawn from the data. Moreover, the expert system can assist in the analyses and control the presentation of results as well.

Current expert systems, however, are characterized by rather narrow knowledge domains. Their expertise is generally very specialized to the task at hand. In order to accomplish the entire range of tasks mentioned in the previous paragraph a collection of systems would probably be required, each one specializing in one area. Perhaps another system could serve as an expert manager for the others.

3.1 Types of Expert Systems

There are a number of organizational schemes for expert systems. Most current expert systems are based upon a type of IF_THEN rule structure. It goes by various names, such as a situation-action or a production rule system. The general technique in all of these rule-based systems is that the recognition of one or more pre-condition states somehow triggers a transformation to a post-condition state. "If we have A, B, and C then we can get to D." Rule-based systems can be implemented in a variety of ways: large case statement structures, list-pattern matching, or as theories in logic which employ resolution to deduce results [Hayes-Roth 1985].

The concept of a knowledge frame [Minsky 1981] employs a data structure into which directly pertinent knowledge about some object or event is concentrated. The frame localizes information on one subject, and provides slots for default values and an interface to other related knowledge frames. For instance a frame could be constructed for a restaurant. Information about seating and the functions of cooking and serving would be included along with some default values like typical menu entries (e.g. hamburgers and fries). A frame for a different restaurant may have similar cooking and serving functions, but the values for seating could be increased and the default menu entries overridden (e.g. by fried rice and egg rolls).

The use of associative (or semantic) networks is another approach [Findler 1979]. Knowledge is arranged into a directed graph structure such that arcs represent relations between various pieces of knowledge which are stored at the graph nodes. The associative network concept may be combined with the rule-based approach if the rules for traversing the network are stored at the nodes.

At this time working with frames or networks is considered difficult because of the problems inherent in standardizing the knowledge structures to gain working efficiencies; however, there is a growing interest in these forms as some of the underlying problems are gradually resolved [Gevarter 1983].

4 Expert Systems Essential For GIS

Any future GIS will necessarily include techniques developed for use in expert systems. Expert systems are of such great utility that we assume that this is a common movement for all forms of database development and to refer to an "expert GIS" would be redundant; however, as mentioned above, currently expert systems have very narrow scopes of interest.

A GIS must be quite expansive in its knowledge domain to provide competent and useful information on geography, cartography, and fundamental spatial relationships.

The wide scope of expertise required for a GIS is too complex for current technologies to handle, however; a GIS as an expert system is, at this time, only a dream. Even a system to handle cartographic output requires knowledge that has not yet been successfully organized.

Despite the size of the problem, however, it is assumed that a comprehensive expert GIS will someday exist. All of the common definitions of a GIS imply that power. When the problems are eventually solved and a system is constructed, it will provide assistance to a diverse user group including private citizens, resource managers, highway and utility planners, and legal and governmental services. To accomplish its mission, a GIS will need to integrate a number of specialized knowledge domains or possibly combine a number of expert systems into one cohesive unit.

4.1 Spatial Organization Experts

A GIS is a spatial information system and it is, therefore, important to consider how best to represent spatial relationships among objects. Humans place special emphasis on the geometric properties of things; they want to know if two objects are near to one another (and usually how near) or if they are touching at a point or along a border line, etc. Perhaps they overlap, intersect, or penetrate. There is an enormous amount of information in even a simple graphic image and a GIS should have access to it all. Unfortunately no one has yet invented a single method for the computer representation of the spatial aspects of objects in a way that is entirely satisfactory. An important research effort is to develop a formal system for reasoning about spatial relations.

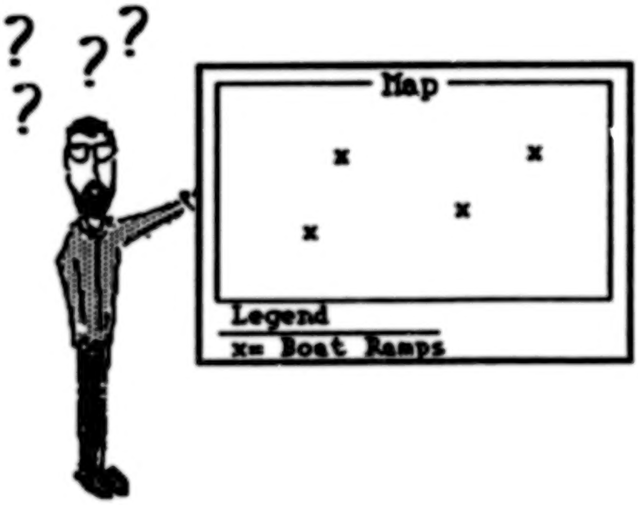
The two major technologies that have evolved in an attempt to treat this problem are raster and vector. Raster is concerned with the direct organization of space and vector is concerned with the organization of objects in space. The differences are not easily reconcilable at the technical level; however, they both address the same problem. Some operations are easier in one system than the other. It is assumed that future systems will find a way to accommodate both raster and vector, but it will probably take another expert system to decide which form or combination of forms is better suited to a particular solution.

4.2 Cartography Experts

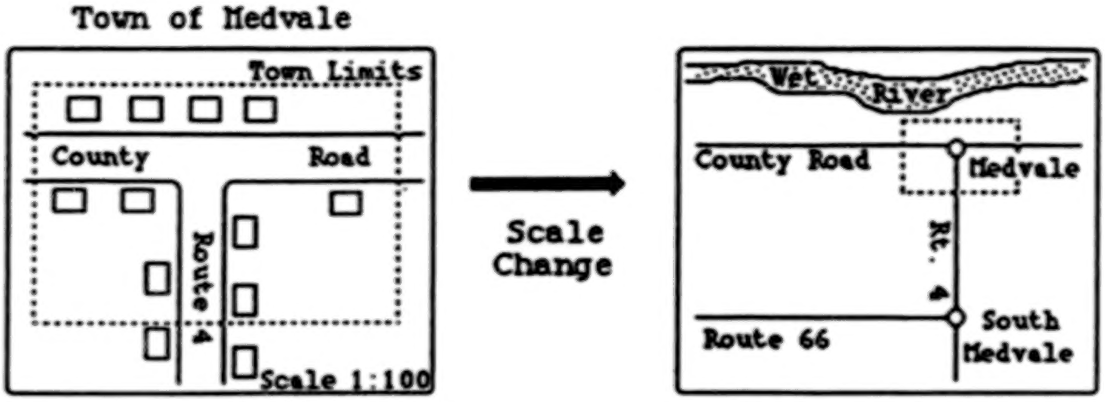
Expert system technology will be crucial to the construction of systems that are to produce cartographic output on demand. Maps that are mass-produced and stocked for subsequent distribution must make compromises in their information content because their audience can not be narrowly defined. The major advantage of a spatial information system over current mapping efforts is that the spatial information system is intended to produce maps that closely match the specific needs of the individual user.

The graphic quality of a map produced by a GIS must remain at an acceptable level, although not necessarily as good as a human cartographer's effort. To accomplish this requires that rather sophisticated decisions be made as to map form and content. Some progress in this area has been made concerning the automatic placement of names of cities and towns, etc. such that overlaps and other conflicts are avoided [Ahn 1984] [Freeman 1984]. At least two levels of problems still remain, however. One is the proper selection of map entities to be displayed and the other is the proper selection and placement of representative symbols for those entities.

A user about to go fishing may ask for a map of all public boat ramps in Maine, but a map that only shows boat ramps will be of little use unless the sea coast and lakes and perhaps a few connecting roads are also shown. Choosing what to display as context for the primary information request requires expert intervention.



At some point during changes in scale a single detailed symbolic representation of some entity or a closely grouped collection of symbols must be either be eliminated or replaced by a simplified or generalized symbol. The entity may be a street map of a town which is changed to a plain colored area showing only major thoroughfares or which is replaced entirely by a small circle, the size of which is proportional to the town's population. The tightly grouped collection of symbols may be small squares representing houses along a street which may simply be eliminated altogether at larger scales. Deciding when and how to modify a symbol in these instances also requires some cartographic expertise.



Scale changes cause symbol changes.

4.3 User Interface Experts

Users must be able to express their needs clearly and simply, in a way that does not require a deep understanding of the technology of the GIS. They should not have to write programs to extract needed results. A system should accept a description of the information required and find ways on its own to produce it with minimal further input. An expert system must make allowances for imprecise information requests and provide assistance in guiding a search for specific information.

In order for this to happen, the system must provide a language that both user and system can tolerate. This language must permit expressions of spatial relationships, such as inside, next-to, overlapping, as well as other, more conventional, queries. For instance, a request to a GIS might be to identify all of the spruce and pine forests within 15 miles of navigable rivers in Washington County, Maine, or determine that piece of land which is zoned as light industry and which is nearest to the town of Orono. It is preferable, but not absolutely necessary, that the language be close to a natural language in form and style.

To respond best to a user's request, a system needs some internal model of the user and his or her current focus. Human responses are somewhat adapted to the perceived audience and a GIS should have similar capabilities. Is the user interested in just an overview or some specific, detailed piece of information? Does he or she want an analysis of the results? Is the user interested in highway construction, utility planning or environmental issues? Does the user have extensive experience in this subject or does he or she need help in finding information or with understanding the information? Is this user color blind? Terminology, verbosity and help levels should all be variables in a GIS-user interface. Progress has been slow in this area [Friedell 1982] and much more work needs to be done.

4.4 Data Quality Experts

The quality of information product of a GIS is intimately associated with its reliability in decision-making. To make effective use of a GIS a user should know what the level of confidence to place in the results of the system. The expression "garbage in – garbage out" stills holds, but it is also possible to get garbage out even with a respectable input. An information system is a data processing system and all non-trivial computer transformations introduce errors which may compound as they propagate through the system [Hudson 1987]. As an expert in geographic information the system itself may be used to track data quality and provide the following:

- a listing on demand of the statistical confidence level of any piece of data displayed.
- warnings that the transformations employed have compounded errors to the point that the data requested is now useless.
- warnings against the inappropriate use of certain operations. The arithmetic mean of Portland and Bangor should not be computed.
- guidance in the proper collection and quality requirements of primary geographic data such as a suggestion that to obtain an output of a particular precision A requires that the primary input data be at least of precision B.

5 Conclusion

The work done by AI researchers to find general purpose solutions to the problems of getting a computer driven system to act intelligently can often be adapted successfully to the more specialized requirements of a Geographic Information System. Particular methods and tools developed in the field of AI, such as heuristically controlling searches, maintaining a separation between the representation of knowledge and the control of its processing, and giving the GIS the ability to learn in order to acquire new knowledge on its own, are indespensible in the construction of a viable GIS. The incorporation of expert systems is necessary because of the size and complexity of the problems expected to be encountered in a GIS. Ordinary procedurally oriented programming techniques will be insufficient to accomplish the task.

Acknowledgement

Thanks to Jeff Jackson who made the illustrations.

References

[Ahn 1984] J.K. Ahn. Automatic Map Name Placement. Technical Report IPL-TR-063, Image Processing Laboratory, Rensselaer Polytechnic Institute, Troy (NY), 1984.

[Friedell 1982] M. Friedell et al. Context-Sensitive, Graphic Presentation of Information. In: D. Bergeron, editor, Proceedings SIGGRAPH '82, ACM Computer Graphics, Boston (MA), July 1982.

[Freeman 1984] H. Freeman and J. Ahn. AUTONAP—An Expert System for Automatic Map Name Placement. In: International Symposium on Spatial Data Handling, Zurich (Switzerland), August 1984.

[Findler 1979] N.V. Findler. Associative Networks. Academic Press Inc., Orlando (FL), 1979.

[Frank 1984] A. Frank. Extending a Database with Prolog. In: L. Kerschberg, editor, Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island (SC), October 1984.

[Glick 1985] B. Glick et al. Hybrid Knowledge Representation for a Geographic INformation System. In: Seventh International Symposium on Computer-Assisted Cartography, Washington (DC), 1985.

[Gevarter 1983] W.B. Gevarter. An Overview of Artificial Intelligence and Robotics. Technical Report, NASA Technical Memorandum 85836, Washington (DC), 1983.

[Hayes-Roth 1985] F. Hayes-Roth. Rule-Based Systems. Communications of the ACM, 28(9), September 1985.

[Hudson 1987] D. Hudson. Data Quality in GIS. Technical Report, Surveying Engineering Program, University of Maine, Orono (ME), 1987.

[Kowalski 1979] R. Kowalski. Logic for Problem Solving. Elsevier Science Publishing Co., New York (NY), 1979.

[Minsky 1981] M. Minsky. A Framework for Representing Knowledge. In: J. Haugeland, editor, Mind Design, MIT Press, Cambridge (MA), 1981.

[Moore 1982] R.C. Moore. The Role of Logic in Knowledge Representation and Commonsense Reasoning. In: AAAI-82, Pittsburgh (PA), 1982.

[Newell 1981] A. Newell. The Knowledge Level. AI Magazine, 2(2), 1981.

[Nilsson 1980] N.J. Nilsson. Principles of Artificial Intelligence. Tioga Publishing, Palo Alto (CA), 1980.

[Robinson 1987] V.B. Robinson and R.N. Wong. Acquiring Approximate Representations of Some Spatial Relations. In: N.R. Chrisman, editor, Eighth International Symposium on Computer-Assisted Cartography, Baltimore (MD), 1987.

[Smith 1986] T. Smith et al. KBGIS-II: A Knowledge-Based Geographic Information System. Technical Report TRCS86-13, Dept. of Computer Science, University of California, Santa Barbara (CA), May 1986.

[Smith 1984] T. Smith and M. Pazner. Knowledge-Based Control of Search and Learning in a Large-Scale GIS. In: International Symposium on Spatial Data Handling, Zurich (Switzerland), August 1984.

[Winograd 1975] T. Winograd. Frame Representations and the Declarative/Procedural Controversy. In: D.G.Brobow and A.M. Collins, editors, Representation and Understanding, Academic Press, New York (NY), 1975.

BLANK PAGE

Suitability of Knowledge Representation for GIS

Piotr Jankowski
Department of Geography DP 10
University of Washington
Seattle, WA 98195
Piotr@uwacdc

ABSTRACT

The choice of a knowledge representation formalism is a dilemma with far reaching consequences on the performance of a knowledge-based information system. Although many formalisms have been proposed, the lack of a formal metric of their appropriateness makes the choice a part of the art of using AI-related techniques. It is suggested that the subjectivity of implementing a knowledge representation scheme can be diminished by an analysis of knowledge structure which can indicate a suitability of formalism for representing the external world. In this paper the framework for such an analysis is outlined and the role of generic knowledge types as the suitability indicators is discussed.

1. INTRODUCTION

Information representation has been one of the crucial issues in designing and implementing information systems. The nature of the problem lies in encoding a body of information in a structure so that the information can be processed into a form that makes a solution available. Hence, the notion of the general representation problem can be approximated by the following equation:

$$\textit{Representation} = \textit{Information Structure} + \textit{Access}$$

The representation consists of a symbolic structure and a mechanism to access the information encoded in the structure.

In research on geographic information systems the problem of representation has been identified with data primitives and data structures appropriate for the GIS-related data types. Two data types can be qualified as being GIS-related: locational data and thematic data. Locational data, at the primitives level, have been traditionally expressed by geometric objects: point, line, polygon, and volume. Numerous data structures centered around one or more of the cartographic primitives have been suggested for representation of the cartographic data (Clarke, 1986). Each of these structures exhibits a different representation scheme in terms of storage, access, and different computational capabilities which can be characterized along the tradeoff: computational efficiency versus storage efficiency. Thematic data used in GIS have been widely represented by data structures designed for data base management systems (hierarchical, network, relational).

The use of information in conventional GISs as data only (information about specific instances) brings about an important consequence for the functional capabilities of the system. Only a fraction of knowable reality is represented in the system. This fraction consists of the information about reality which can be expressed at the instance level - in a form of data. Yet the system does not have any information or very little at best about objects' properties which cannot be captured in form of data about relations binding objects and concepts explaining them. This portion of information about knowable reality, which a conventional GIS lacks, has a more general character and therefore can be called knowledge. Knowledge, in contrast to data, does not deal with specific instances but it rather involves the information about general concepts and relations.

There are at least two readily identifiable advantages of having the data in a geographic information system supplemented by knowledge. Firstly, more information relevant to the problem domain can be encoded in the system which enhances its potential functionality. Secondly, the system can draw inferences and perform reasoning upon the data, providing it is equipped in appropriate mechanisms, which would normally require a human expert. The price paid for these improvements is an increased complexity and a lack of "paved" and clearly marked roads in designing a system.

The potential advantages of including problem domain knowledge in the GISs have been recognized (Coulson et al. 1984, Kubo 1986, Peuquet 1986, Robinson et al. 1986, Smith 1984). Efforts in this new area of GIS research have concentrated on the design and implementation of prototypical Knowledge Based Geographic Information Systems (KBGIS) also called Intelligent Geographic Information Systems (IGIS). In KBGIS the four basic modules of a conventional GIS (data input, storage and retrieval, manipulation and analysis, data output) are supplemented by a domain knowledge module (knowledge base + inference mechanism). Although the content and the scope of knowledge bases reported in the KBGIS projects differ, there seems to be a consensus among the researchers that the system should exhibit the knowledge of spatial concepts, relationships between concepts, and a knowledge permitting decision-making capabilities (Kubo 1986, Smith 1984). These knowledge requirements are necessary conditions for the realization of potential functions of KBGIS like:

- interpretation of analysis results,
- user-directed guidance of efficient applications of the system,

- knowledge acquisition and learning,
- checking for consistency and reliability of information,
- handling the incomplete data,
- modeling and hypothesis testing,
- decision-making support.

One of the main issues in designing any knowledge-based system, including KBGIS, has been how to represent knowledge domain. The importance of knowledge representation is that the efficiency of finding a solution to the problem depends greatly on the scheme in which knowledge is stored and accessed. Although at first glance it seems like the issue can be put into the category of technicalities the choice of knowledge representation scheme, has profound consequences for the design, development, and efficiency of a knowledge-based system. The efficiency of finding a solution is a decisive factor in evaluating the usefulness of the knowledge-based system especially as a decision support tool or a problem solver.

Despite the fact that knowledge representation has been one of the central research problems in artificial intelligence there is no solid theory of the subject. This fact makes the choice of knowledge representation scheme a part of the art of using AI-related techniques. Yet the subjectivity of implementing a knowledge representation scheme can be diminished by a suitability analysis of formalisms for representing problem domain. It seems to us important that such an analysis should precede the process of designing a KBGIS especially if the system implements AI-developed knowledge representation schemes.

In this paper the suitability of knowledge representation schemes is analyzed at the conceptual level. Indicators are examined which can be used in evaluating the suitability of knowledge representation schemes for GIS.

2. LEVELS OF ANALYSIS

A point of view shared by many researches is that the choice of representation scheme does not effect substantially the reasoning abilities of the knowledge based system. Any knowledge representation scheme or representation language which is equipped with the truth theory and which can express the knowledge of the system in the general propositional form can also make inferences upon the system's knowledge. From the reductionist point of view every knowledge domain could even be represented by the most general representation language - first order predicate logic. There is however, a basic tradeoff between the expressiveness of a representation scheme and its computational efficiency which underlies different knowledge representation schemes (Levesque, Brachman 1985). It has been shown empirically that although the expressive power of first-order logic is high its computational efficiency falls far below that of more restricted representation schemes like production systems or frame systems (Niwa, Sasaki, Ikara 1983). There are two important implications of this tradeoff. The first one is that the knowledge acquired for a system which has to reason efficiently ought to comply with restrictions regarding the incompleteness and the lack of structure. The second one is that forcing a given domain knowledge into an inappropriate representation scheme (e.g. trying to represent a poorly formalized domain in a specialized frame language) can significantly reduce the system's efficiency of finding a solution to a problem. The important question arising here is how to choose a suitable representation scheme for a given problem knowledge and what are the selection criteria for making the choice.

A key issue in analyzing the suitability of a scheme for representing knowledge is the level of analysis. The analysis conducted at the level of symbolic structures used for knowledge representation deals with the question of "how to say it". In other words a knowledge domain is taken as a given and one looks into possible ways of representing it. Although this level of analysis seems to be important from the system design point of view it focuses mainly on techniques and tools and it may result in neglecting the content of the problem.

The higher level of analysis which deals with the question "what it says" is the conceptual level (or knowledge level in Allen Newell's words, Newell 1982). It seems logical that before trying to analyze various ways of "how to say it" one would examine the content in terms of "what it says". At the conceptual level one looks at the content of what is the problem knowledge in terms of what it says about the problem and what it takes to solve it. Such a notion of knowledge has a functional character. It provides us with the general account of facts, definitions, and actions resulting from them which can be taken to attain the goals. By analyzing the content of the problem knowledge we can derive the knowledge chunks in a form which has a known semantics (e.g. general propositional statements). These pieces of knowledge have a certain underlying structure. Uncovering this structure is essential for representing knowledge at the symbol level. In fact, the organization of knowledge at the conceptual (knowledge) level can be mapped into the proper knowledge structure and processing mechanism at the symbol level. Model of knowledge organization at the knowledge level can be perceived as a graph in which nodes express concepts and arcs denote relations between them (Sowa 1984). Relations as well as other structural knowledge elements can be classified into general categories which are common to various knowledge domains including domains amenable to geographic information systems and hence, should be called generic knowledge types.

3. GENERIC KNOWLEDGE TYPES

There are at least five knowledge types which can be identified (Wiederhold et al.1986):

- factual knowledge,
- operational knowledge,
- categorical knowledge,
- definitional knowledge,
- causal knowledge.

In the simplest case a model of knowledge structure would consist of facts and operations. Facts have a propositional form of statements and assertions about objects' attributes. Operational knowledge focuses on advice for reasoning about facts. It can be thought of as a knowledge that deals implicitly with the combinatorics of the problem domain. In a model of shallow knowledge whose structure describes associations and correlations, operational knowledge is imbedded in associative and functional relations.

Categorical, definitional, and causal knowledge types are pertinent to deep knowledge structures which include causative relations. Categorical type underlies a hierarchical organization of knowledge. The important function of categorization of the knowledge is that it provides guidelines for a decomposition of a larger problem into a complete set of sub-problems. The hierarchy resulting from the problem decomposition supports inheritance which eliminates redundancies in representing a complete problem knowledge. Definitional knowledge provides definitions for object and attribute abstractions and it can also provide means of computation and default values for missing attributes. The essence of causal knowledge is how one state effects another state. Causal knowledge can be perceived as consisting of the linkages and dependencies among the elements of hierarchy. These relationships can be described by a set of features (e.g. intensity, frequency, direction, time).

The characteristics of knowledge types indicate some clear preferences regarding the suitability of AI-developed knowledge representation schemes. Hence, the knowledge types can be treated as indicators in analyzing the suitability of using certain schemes for representing problem domain.

In order to use knowledge types as suitability indicators one has to start with formulating a conceptual model of problem knowledge. As mentioned above, such a model should express the structure of the problem knowledge. One of the possible approaches to specifying a conceptual model is to depict it in a form of a graph with nodes representing concepts and arcs denoting relations. Through the analysis of such a model the general conceptual structure elements (concepts, relations) can be decomposed into subconcepts, key variables, facts, and specific relations (associative, functional, and causal). These specific elements can be categorized as the constituents of generic knowledge types. By enumerating the structural elements in terms of knowledge types one can evaluate which representation scheme is suitable to represent the structure of a given problem knowledge.

Depending on the structure of problem knowledge the generic knowledge types can be found as underlying domains amenable to geographic information systems. Factual and operational knowledge can be generally well represented by conditional statements of the type: " IF this fact is true THEN draw this conclusion". Those statements provide the basic structure for rule-based representation schemes. Rules are suitable for representing shallow knowledge. Factual and operational knowledge are necessary and often sufficient for a system to perform diagnostic and management functions like: answering user's queries, analysis of results, user-directed guidance of the efficient system applications. A suitable paradigm for representing the factual and operational knowledge is also the first order logic (FOL). The FOL is generally suitable for representing domains whose knowledge structure is not or cannot be well described. However the usefulness of FOL to represent poorly formalized domains is based on the tradeoff: expressiveness versus computational efficiency. A system based on FOL is often computationally inefficient.

The categorical, definitional, and causal knowledge types are amenable to deep knowledge. They underlie the domains for which a knowledge structure is well developed (formalized knowledge domains). These knowledge types can be suitably represented in semantic networks and frames. The frame-based scheme is especially well suited to represent inheritance - a property which can be derived from categorical knowledge, definitional knowledge in form of defaults, and causal relationships. Other mechanisms provided by frame-based schemes (e.g. slots, demons) also provide for representation of factual and operational knowledge types. However, the frame-based systems are in general more difficult to design and to implement than the rule-based systems. Causal, definitional, and categorical knowledge types are suitable to support learning, explanatory, and data reliability checking functions of a system.

Some potential applications of GIS implicate the simultaneous occurrence of all generic knowledge types in one knowledge domain. In cases like that there are often more than one suitable representation schemes which can be considered. Hybrid schemes involving more than one representational paradigm seem to be a useful solution. Mixing frames with rules one can represent shallow and deep knowledge on one hybrid representation scheme. A hybrid scheme in which frames are linked with simulation models can be used to express causal knowledge and the data (Ziegler 1987).

SUMMARY

The choice of knowledge representation scheme effects the design, implementation, and the efficiency of reasoning in a knowledge-based system. A representation scheme should be chosen as a result of the analysis of problem knowledge at the conceptual level. At this level of abstraction the model of structure underlying a knowledge domain can be derived. Such a model can be depicted in the form of a graph. The elements of the conceptual structure can be categorized as generic knowledge types. These types underlie many knowledge domains amenable to GIS. Since the knowledge types reveal some clear preferences in regard to standard knowledge representation schemes, they can be used as indicators of the suitability of these schemes to represent a problem-oriented knowledge.

REFERENCES

Clark, K. 1986: Advances in geographic information systems. *Comput. Environ. Urban Systems*, vol.10, no.3/4:175-184.

Coulson, L., Folse, J., Loh, D. 1987: Artificial intelligence and natural resource management. *Science*, vol.237:262-267.

Kubo, S. 1986: The basic scheme of TRINITY: a GIS with intelligence. *Proc. 2nd International Symposium On Spatial Data Handling*, Seattle, pp.363-374.

Levesque, H., Brachman, R. 1985: A fundamental tradeoff in knowledge representation. In *Readings in Knowledge Representation,* eds. R. Brachman, and H. Levesque, pp.42-70. Los Altos: Morgan Kaufmann Publishers, Inc.

Newell, A. 1982: The knowledge level. *Artificial Intelligence*, vol.18, no.1:87-127.

Niwa, K., Saski, K., Ihara, H. 1983: An experimental comparison of knowledge representation schemes. *AI Magazine*, vol.5, no.2:29-36.

Peuquet, D. 1984: Data structures for a knowledge-based geographic information systems. *Proc. International Symposium On Spatial Data Handling*, Zurich, pp.372-391.

Robinson, V., Frank, A., Blaze, M. 1986: An assessment of expert systems applied to problems in geographic information systems. Paper presented at the ASCE Specialty Conference on Integrated Geographic Information Systems, San Diego.

Smith, T. 1984: Artificial intelligence and its applicability to geographical problem solving. *The Professional Geographer,* vol.36, no.2:147-158.

Wiederhold, G., Blum, R., Walker, M. 1986: An integration of knowledge and data representation. In *On Knowledge Base Management Systems*, eds. M.Brodie, and J. Mylopoulos, pp.431-443. New York: Springer-Verlag.

Ziegler, B. 1987: Knowledge representation from Newton to Minsky and beyond. *Applied Artificial Intelligence*, vol.1, no.1:87-107.

MULTI-COMPONENT OBJECT SEARCH USING SPATIAL CONSTRAINT PROPAGATION

Sudhakar Menon
Terence R. Smith
University of California, Santa Barbara, CA 93106

Abstract

This paper discusses the task of searching for multi-component spatial objects in a geographic database, when the relationships between components are only implicitly stored. While the general problem of spatial constraint satisfaction is shown to be NP-Hard, multi-component objects of practical significance are extracted through the efficient propagation of spatial constraints in the database. Such a strategy is shown to prune space and limit the portions of the database that are searched. The specific data structures and procedures used in KBGIS-II, a GIS implemented at the University of California, Santa Barbara are discussed. A search for potential landslide sites in the Black Hills, defined using an object model with five components, is used to illustrate the concepts described.

1. Introduction

Many spatial objects of interest to the user of a GIS may be represented using models that contain multiple parts or components linked by specified spatial relationships. Each component may be required to satisfy various attribute constraints. Such multi-component spatial objects may be explicitly represented in a GIS. For example, the locations of a spatial object with n components may be represented using an n-ary relation defined on the sets of locations for each component. In general however, a flexible GIS cannot rely on the explicit representation and storage of all spatial relationships and objects that may be of significance to a potential user.

In this paper we study the computational task of **dynamically** assembling together locations of multi-component objects given a definition of the object in terms of its parts. The individual components may be either explicitly stored in object based (vector) databases or implicitly stored in location based (raster) databases. The multi-component spatial search problem is first formally defined and its complexity analyzed. Algorithms for search based on the concept of spatial constraint propagation are then presented. Spatial constraint propagation is used to replace the explicit checking of spatial constraints by geometric searching within constrained areas of the database. Following a brief review of data structures suitable for constraint propagation, an actual GIS implementation of multi-component spatial search is described using a search for potential landslide sites in a mountainous area as an example.

2. The Multi-Component Spatial Search Problem

The Multi-Component Spatial Search Problem (MCSSP) may be formally defined as follows :

Let O be a spatial object with n sub-objects $O_1, \ldots, O_n$. A sub-object may be be subject to unary constraints and any pair of subobjects to binary constraints. Let $v_1, v_2, \ldots, v_n$ be variables that take as values the **locations** of each of the n sub-objects. Each sub-object has a finite number of instances in the spatial database, with a unique location corresponding to each instance. A location of a sub-object may be a connected region whose boundary is a polygon, a piecewise linear curve or a point in the plane. Each variable v_i can take on values from a finite domain D_i which represents the set of all locations of the sub-object O_i in the spatial database. The domains D_i are not required to be disjoint.

A **location-set** for the object O is a set of locations, one for each sub-object of O, such that all unary and binary constraints are satisfied. Examples of unary constraints include specifications on the size, shape or attribute values associated with the locations of a sub-object. Examples of binary constraints include specifications on the distance between the locations of two sub-objects or the relative direction between these locations. A location-set is thus a set of n locations :

$\{x_1, \cdots, x_n \mid P_1(x_1) \wedge \cdots \wedge P_n(x_n) \wedge P_{12}(x_1, x_2) \wedge \cdots \wedge P_{n-1,n}(x_{n-1}, x_n)\}$

where $x_i \varepsilon D_i$ for $i=1, \cdots, n$. In the above expression predicate P_i is used to represent the conjunction of all unary constraints on the location variable v_i, predicate P_{ij} is used to represent the conjunction of all binary constraints on the location variables v_i and v_j, for $i, j=1, \cdots, n$.

The object O may be assigned a unique location x corresponding to each location-set , where $x=f(x_1, x_2, \cdots, x_n)$ The function f may be different for different spatial objects. As an example f may return the centroid of its arguments, the convex hull of its arguments, or project a specific location from its set of arguments. The MCSSP consists of determining all location-sets for the object O, given the domains of each of the n sub-objects.

3. The Computational Complexity of Multi-Component Spatial Search

A problem is considered computationally easy if it permits a solution that takes time polynomial in the size of the input. Multi-component spatial search is a computationally hard problem. The NP class of problems includes all those problems for which a solution, if exhibited, could be **verified** in polynomial time. A problem in the class NP is NP-Complete (NPC) if **any** problem in the class NP can be reduced to it in polynomial time. This implies that the existence of a polynomial time algorithm for any NPC problem would imply the existence of a polynomial time algorithm for all other problems in the class NP. It is conjectured that no polynomial time algorithms exist for problems in NPC. Examples of NPC problems include the Traveling Salesman Problem, Graph Coloring and the Sub-graph Isomorphism Problem. The time taken by all known algorithms for these problems grows exponentially as the size of the input to the problem grows.

3.1. Multi-Component Spatial Search is NP-Complete

We show that the spatial search problem is NPC, by exhibiting a polynomial time transformation of graph 3 colorability, a known problem in NPC (Garey and Johnson, 79) to the spatial search problem. The graph 3 colorability problem is defined as follows :

Instance : a graph G = (V, E), and 3 colors = {r, b, g}

Question : Is there a function f that assigns colors to the nodes of G so that $f(i) \neq f(j)$ if $(i,j) \in E$.

All known algorithms for graph coloring have costs that grow exponentially as the number of nodes in the graph is increased. Given an instance of graph 3 colorability as defined above we may construct the following corresponding instance of multi-component spatial search. The object O corresponds to the graph G. O has n sub-objects $O_1, \ldots, O_n$, where $|V| = n$. The locations of each sub-object are restricted to points (x, y) in the plane. There are no unary constraints on the sub-objects. There is a binary constraint P_{ij} between the location variables v_i and v_j iff there is an edge (i,j) $\in$ E. Each binary constraint P_{ij} is of the form distance(v_i, v_j) > 0. Thus $P_{ij}((x_i,y_i),(x_j,y_j))$ is true iff $(x_i, y_i) \neq (x_j, y_j)$. We select any three distinct points in the plane corresponding to the colors r, g and b. The domain of each of the n subobjects is identical and corresponds to this set of three points. Then the graph G is 3 colorable iff there is a location-set corresponding to the object O. The proof of the above claim is obvious. Graph 3 colorability is thus polynomially transformable to spatial search. The spatial search problem is thus NPC. The above argument can be applied to both region and line based objects. In reality the total number of solutions to the MCSSP may grow exponentially as the number of sub-objects increase. It is therefore not possible to even verify all solutions to a problem in polynomial time. The problem of determining the set of **all** solutions to the MCSSP is thus NP-HARD (Garey and Johnson, 79).

4. Characteristics of Multi-Component Spatial Search

The Multi-Component Spatial Search Problem is a special case of the Constraint Satisfaction Problem (CSP). The CSP is a well known combinatorial problem in Artificial Intelligence and arises in several domains including Computer Vision. The input to the CSP is a set of n variables and a set of relations each involving a subset of variables. Each variable can take values from an associated domain. The CSP requires the finding of all possible n-tuples of values such that each n-tuple is an instantiation of the n variables satisfying all the relations. The set of variables and the binary constraints between them in a given CSP can be represented using a constraint graph, in which each node i represents a variable v_i, and an edge between two nodes represents a symmetric binary constraint between the two variables. At most a single edge joins two nodes even if more than one binary constraint is present in the given CSP.

The MCSSP is seen to be a special case of the CSP with the following properties

(i) The constraints between variables are spatial in nature.

(ii) The values that variables can take are locations in the plane.

Algorithms for spatial search can thus seek to exploit the spatial nature of both the domain and constraints by using appropriate data structures and retrieval algorithms.

The instances of the MCSSP commonly encountered in GIS applications possess the following additional features :

(a) : The constraint graphs representing spatial objects are usually sparse. This is in contrast to widely studied instances of the general CSP such as the 8 - Queens problem in which the constraint graph is a clique with constraints between every pair of variables.

(b) : Objects of practical interest have a relatively small number of sub-objects. Because the MCSSP is NPC, search times may be expected to increase exponentially with the size of the constraint graph. From a pragmatic viewpoint however most instances encountered will be small.

(c) : The sizes of the variable domains are very large in the MCSSP. In a spatial database each sub-object may have a very large number of examples (locations). As an example there are approximately 215 first order streams in a 512 x 512 window of a dataset covering the Black Hills, S. Dakota, used in this study. This is in contrast to problems such as the 8 - Queens problem where each of the 8 variables (rows) can take on at most 8 values (columns). For efficient spatial search it is therefore necessary to reduce the complexity of search as a function of domain size.

(d) : For any arc (i, j) in the constraint graph for a MCSSP, the average number of values in D_j that are consistent with any selected value from D_i is expected to be a small percentage of the total number of values in D_j. This will of course depend on the specific constraint involved. As an example, of the 215 first order streams described above, there will only be a small fraction that are near any specific ridgeline. This may be described by saying that the degree of constraint in spatial search is high. This is a characteristic shared by the MCSSP with instances of the sub-graph isomorphism problem on large sparse graphs, but not with the 8-Queens problem.

(e) : Spatial constraints are not independent. The existence of a spatial constraint between sub-objects O_1 and O_2, and of another spatial constraint between sub-objects O_2 and O_3, implies the existence of a spatial constraint between O_1 and O_3. If this constraint can be inferred, it may be possible to exploit it during search.

5. A Review of Algorithms for the CSP

This section reviews algorithms for the CSP as background to the description of algorithms for MCSSP. The state space for the CSP may be viewed as a tree with n levels corresponding to the n variables. Each leaf of the tree corresponds to a member of the Cartesian product of the n variable domains. Backtracking is the basic tree search algorithm, and corresponds to a depth first search of the state space. Variables are sequentially assigned with values from their domains. If no value in the domain of a variable is found to be consistent with the assignments made to earlier variables, the procedure backtracks to the preceding variable. The time taken by backtracking to find a solution for the general CSP is exponential in the number of input variables in both the **worst** case and in the **average** case (Knuth, 75, Brown and Purdom, 81).

Backtracking can be highly inefficient and exhibit a class of behavior referred to as thrashing. Mackworth (77) defined three criteria on the constraint graph, referred to as node, arc and path consistency, that serve to minimize thrashing.
Node i is node consistent if and only if any value x in D_i satisfies $P_i(x)$.
Arc (i, j) is arc consistent if and only if for any value x in D_i there is a value y in D_j such that $P_j(y)$ and $P_{ij}(x,y)$ hold.
Path consistency extends the concept of Arc consistency to paths of length greater than two in the constraint graph. Algorithms to achieve consistency may be applied before backtracking, to minimize thrashing, and are discussed in Mackworth (77) and Mackworth and Freuder (85). A well known example of an arc-consistency algorithm is the Waltz Filtering algorithm (Waltz, 75).

Other alternative improvements to the backtracking algorithm have been suggested. One basic idea is the dynamic enforcement of an arc consistency condition during the tree search. This idea was used by Ullman (76) and further developed by McGregor (79) in constructing algorithms for the sub-graph isomorphism problem. One of the most successful variants (Haralick and Elliott, 80) of backtracking is Forward Checking. Forward Checking dynamically imposes an arc consistency condition on the constraint graph during the tree search by looking ahead to ensure that each future variable has at least one value that is compatible with the labels currently held by past and current variables.

6. A Spatial Constraint Propagation Algorithm for the MCSSP

In the general CSP the domain of each variable is represented as an ordered set or list. During backtracking tree search, the values for any variable are tried sequentially. Constraints are explicitly computed in order to check whether the value selected from the domain satisfies the constraints on the variable. No attempt is made to exploit any underlying organization inherent in the variable domains.

The spatial nature of the domain and of the constraints in the MCSSP can be exploited in reducing the computational cost of backtracking while searching for examples of a multi-component object. Spatial constraint propagation can be used to replace the explicit checking of constraints during backtracking by geometric searching within constrained areas of the database.

The process is illustrated using a simple instance of MCSSP, illustrated in Fig. 1. The object to be searched for is modeled by 2 sub-objects O_1 and O_2. The domain of each sub-object is a set of point locations. The figure shows the spatial configuration of the point examples of the two sub-objects. The point locations of O_1 and O_2, are represented by a's and x's respectively. There is a single distance constraint between O_1 and O_2, of the form $distance(a_i,x_i)<R$, where a_i and x_i are locations for O_1 and O_2 respectively, and R is the radius of the circles shown in the figure.

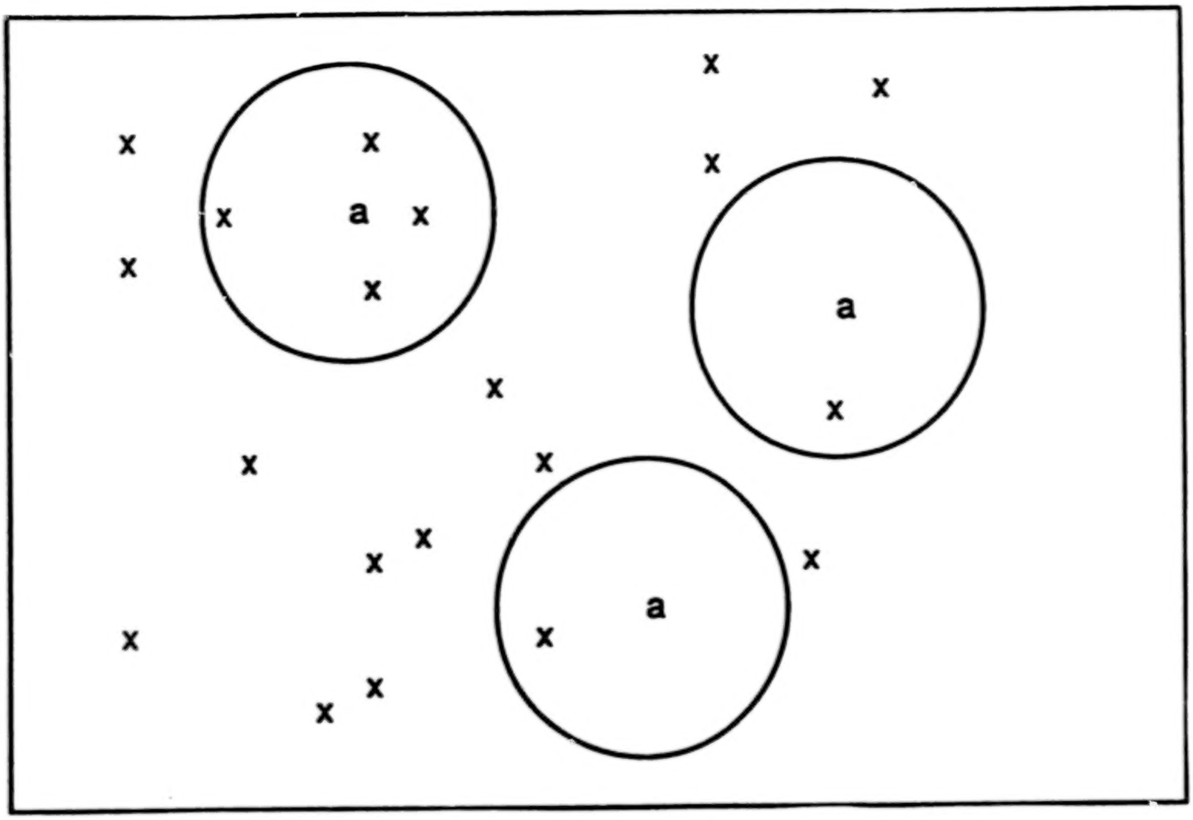


Figure 1. Spatial Constraint Propagation

If there are a_1 and a_2 examples of each sub-object, then using classical backtracking and given a representation of each domain as a list, all consistent solutions can be found by checking all pairs of distances in $O(a_1a_2)$. In the present instance 60 distance computations will need to be carried out.

As an alternative to checking each such distance the spatial constraint may be propagated from each location of O_1. For any given location of O_1 we can construct a circular window of radius d around the location, as shown in the figure. The task of finding locations of O_2 consistent with this location of O_1 can then be reduced to the task of searching for examples of O_2 within the circular window. If an appropriate data structure is used to store the locations of O_2 then this task may be accomplished in sub-linear time, such as $O(a_2{}^{1/2})$. The overall task of constraint satisfaction could then be accomplished in time $O(a_1a_2{}^{1/2})$. In the present instance, 3 constrained area descriptions would need to be generated, followed by 3 searches within these areas.

In practice, depending on the geometric nature of object locations and of the propagated constraints, the constrained search window may have an arbitrarily complex shape, which may not be efficiently searchable. It may then be necessary to search within a larger area which is more efficiently accessed under the geometric data structure selected. Principles of filtering search may then be applied and explicit constraint checking used to filter out retrieved examples that do not satisfy the required constraints.

Replacing constraint checking by constraint propagation does not remove spatial inconsistency and the possibility of thrashing during backtracking, when searching for objects with a large number of components. Consistency algorithms can be adapted to take advantage of constraint propagation techniques. Forward checking, which dynamically applies an arc consistency criteria during backtracking, can be directly incorporated into multi-component spatial search. Each forward check involving the current variable and future variables can be accomplished by propagating the relevant geometric constraints. Additional implicit spatial constraints that are inferred between sub-objects can be used to aid in this task.

7. Data Structures for Spatial Constraint Propagation

Spatial Constraint Propagation requires the use of data structures that allow efficient retrieval of the subset of locations within a constrained window from the total set of locations in the database. This retrieval must be done without the overhead of examining each member of the total set. Most GIS support spatial indexing capabilities that allow efficient retrieval of objects within rectangular query regions.

Points : Data structures that support rectangular query windows include the multidimensional binary tree or k - d tree (Bentley , 75) and the layered range tree (Willard, 78). Other data structures have been specialized for particular query window shapes such as circles of fixed radii (Chazelle and Edelsbrunner, 84), annular query regions (Cole and Yap, 84) and K-sided polygonal query regions Willard (82).

Regions and Lines :

Several methods have been developed for locationally indexing object based representations of regions and lines to allow for fast retrieval within query rectangles. Abel and Smith (83) use a locational key based on the quadtree decomposition to retrieve those members of a set of rectangles that edge-intersect, are enclosed in or contain a given query rectangle. Object based representations of lines and regions are spatially indexed by assigning each feature a key based on its minimum bounding rectangle. Matsuyama et. al. (84) use a k-d tree to spatially index objects in a GIS, based on their centroids. Modifications to the data structure are required to ensure that objects are not missed because of the use of the centroid to compress all locational reference information. Blackwell (87) describes the use of R-trees, which use a hierarchy of minimum bounding rectangles, to construct fast spatial indexes that support rectangular query regions for both region and line data in a GIS.

Location Based Approaches In a location based data model, objects are not represented, rather each location unit (cell, pixel) has an object attribute or property. Point data are not usually stored in location based models, though binary arrays may be used for dense distributions. Region and Line Data may be stored directly in raster form or using a number of pointer-based and key-based quadtree data structures (Samet, 84). Retrieving objects from such a locational representation requires the aggregation of location units into region and linear features using connected component labeling and line following algorithms respectively. This aggregation represents a search overhead associated with extracting object based information from a locational representation. The most easily queried window shape depends on the tesselation used. Most location based data structures use a square or rectangular tesselation model and allow efficient search only within rectangular areas. Recursive partitioning schemes such as the quadtree impose additional address generation or access costs when searching rectangular areas that do not correspond to quadrants generated in the partitioning process. Other query regions such as circles or annuli carry higher search costs associated with locational address generation.

8. Ordering Multi-Component Object Search

In order to use spatial constraint propagation techniques during backtracking, it is necessary to decide on a sequential order in which sub-objects will be searched for. If explicit constraint checking is to be used then all examples of each sub-object are retrieved before backtracking is begun. Again an order must be established in which variables (sub-objects) will be instantiated with values from their (explicit) domains. The order selected can influence the computation path and attendant cost in both schemes.

Several criteria may be been used to derive an ordering :

(1) : Sub-objects should be searched for (or instantiated) in order of domain size, smallest first. The benefit of this approach for spatial constraint propagation is immediately apparent. In the case of Fig. 1, if search were anchored on the x's then 20 constrained searches instead of 3 would result. For explicit constraint checking schemes with more than two variables, the benefit of such a strategy is that the pruning of a variable value at a high level in the search tree results in the largest possible sub-tree being eliminated from consideration. The information on relative frequency is required prior to sub-object ordering and actual search. A GIS can store data on the number of examples of different sub-objects for each section of the database based on either preprocessed or heuristic information. In the latter case approximate average case information on the relative frequency of occurrence of different objects can be stored. A strategy where all sub-objects are completely searched for at the highest level of detail in order to discover their numbers in the search window before the application of spatial constraint propagation is self-defeating. This latter approach can however be used with explicit constraint checking, since constraint verification is always preceded by complete sub-object retrieval.

(2) : Variables may be ordered on the basis of the structure of the underlying constraint graph, in order to increase the level of consistency (Freuder, 82). Such an order may conflict with the order dictated by the size of the domains.

(3) : If the complexities of geometric search differ for the different sub-objects then it may pay to search for objects in the order of difficulty, most difficult last. Sub-objects later on in the sequence will be searched for only if all the constraints between earlier sub-objects are satisfied. Complex sub-objects will thus be searched for in relatively few areas of the database when using constraint propagation, compared to their total

number of occurrences.

9. The Computational Cost of Spatial Constraint Propagation

A comparison between the computational costs of search using spatial constraint propagation and search using explicit constraint checking depends on a number of factors :

(i) The cost of computing an explicit spatial constraint. This cost varies with the spatial relationship involved, the degree of approximation acceptable and the representation used for the geometric objects. For example, there is an order of magnitude difference between the cost of computing the minimum distance between two regions represented as linear quadtrees and the cost of computing the distance between their centroids.

(ii) The cost of computing a constrained search area given an anchor location, the name of the spatial constraint and the desired value of the constraint. In most cases a computationally fast, conservative approximation to the actual constrained area may be used if filtering is used after retrieval.

(iii) The cost of searching a constrained area for locations of an object. This cost is dependent on the data structure used to store the object locations, and the shape of the area to be searched. For example, the exact retrieval of the set of regions lying within 1000 and 2000 units of another region requires search within an approximately annular query window. Most GIS data structures will not support fast retrieval from within such a query region. If this particular query region is approximated by its minimum containing rectangle, then a large amount of filtering will be required. On the other hand constraints such as embodied in the CONTAINS relationship can be effectively propagated using minimum bounding rectangle approximations to the query window.

(iv) The sub-object ordering used. The ordering selected influences computation costs in both constraint propagation and constraint checking schemes. Constraint propagation, costs unlike explicit constraint checking are sensitive to sub-object ordering even when there are only two sub-objects present, as discussed in the preceding section.

Our experience indicates that spatial constraint propagation is most effective when the degree of constraint imposed by the model is high. The degree of constraint can be roughly defined by the size of a propagated query window in relation to the size of the total database window within which search for the multi-component object is being carried out. The degree of constraint is usually high for large data bases, or when the model involves components that are clustered together in space. Spatial constraint propagation is effective when there are large disparities in the number of examples of different sub-objects present. By anchoring search on objects with fewer examples, considerable savings over explicit constraint checking techniques can be expected.

10. An Implementation supporting Multi-Component Object Search

We use a query for the locations of potential landslide sites in a mountainous area to describe search within KBGIS-II, a system implemented at the University of California, Santa Barbara (Smith et. al., 1987). The spatial search component of KBGIS-II is currently implemented in both COMMON-LISP and C. Two knowledge bases named the Spatial Object Knowledge Base and the Function Knowledge Base, are resident at all times within the LISP environment and are implemented using frames. The Spatial Object Knowledge Base stores definition and feature type information on spatial objects. High level search, backtracking and all functions for propagating and checking spatial constraints are coded in LISP. Actual locations of spatial objects are stored in separate Spatial Databases. These Spatial Databases are queried during multi-component search using search functions written in C. The Spatial Object Knowledge Base also allows the name of a special purpose search function to be associated with a spatial object. If such a function is stored, then locations of the object are assembled during search by using the special purpose function to access the Spatial Databases.

Query Description The following test query formulated by geologists at the U.S.G.S., involves the location of worst case potential landslide sites in an area of the Black Hills, S.Dakota. A worst case potential landslide site is a region which :

has a Northwest Aspect (270 degree - 360 degree)
has a Shallow Slope (< 20 degree)
is located on one of a variety of shales and limestones specified by their codes in the Geology thematic layer
is within 200 feet of the upper end point of a first order stream.
is within 800 to 1000 feet of a topographic ridgeline.
is within 200 feet of the boundary between two specified geologic strata
is within 200 feet of a sharp topographic slope break

Knowledge Base Organization The above query is answered by formulating a spatial object with 5 sub-objects, one of which is the sought for region. The descriptions of these spatial objects, named SLOPE_BREAK, GEOLOGIC_INTERFACE, RIDGE, UPPER_END_STREAM and PROSPECTIVE_SLIDE, are entered into and stored in frames in the Spatial Object Knowledge Base. SLOPE_BREAK, GEOLOGIC_INTERFACE and RIDGE are identified as Linear features. UPPER_END_STREAM and PROSPECTIVE_SLIDE are identified as Region features. A special purpose search function, described later, is stored in the frame for UPPER_END_STREAM. The frame for PROSPECTIVE_SLIDE stores its definition in terms of the values in the Geology, Aspect and Slope layers of the spatial database.

Spatial Database Organization The Spatial Database for Linear features uses a locationally indexed object based data structure that allows querying within rectangular windows. Examples of the spatial objects SLOPE_BREAK, GEOLOGIC_INTERFACE and RIDGE are stored in this database together with other linear features such as FIRST_ORDER_STREAMS. The system provides database creation routines that may be used to build this database. Edge detection and line following procedures may be applied to extract linear features from raster images. The

Spatial Database for Region features uses both location based and object based representations. The location based representation uses bit-mapped multi-colored one-to-four autumnal quadtrees, described elsewhere in this volume (Gao et. al, 87, Menon et. al. 87), to store data on different thematic layers such as Landuse, Geology, Elevation, Slope and Aspect. Examples of region objects extracted from the location based database may be separately stored in an object based data structure. All database search and access routines are coded in C.

Query Processing The user enters the following query formulated in a predicate calculus based language, to initiate search for potential landslide sites :

```
(AND
    ((TYPE O_1) SLOPE_BREAK)
    ((TYPE O_2) GEOLOGIC_INTERFACE)
    ((TYPE O_3) RIDGE)
    ((TYPE O_4) UPPER_END_STREAM)
    ((TYPE O_5) PROSPECTIVE_SLIDE)
    ((DISTANCE O_1 O_2) (0 200))
    ((DISTANCE O_2 O_5) (0 200))
    ((DISTANCE O_3 O_5) (800 1000))
    ((DISTANCE O_4 O_5) (0 200))
    )
```

The high level search routine translates the symbolic definition of a spatial object to be searched for into a semantic network representation of the underlying constraint graph.

If the spatial object consists of a single-component then the high level search routine initiates the appropriate database search routines based upon information stored in the Knowledge Bases. For example the PROSPECTIVE_SLIDE object is searched for by intersecting the Geology, Aspect and Slope layers of the Spatial Database for Region features, within the specified constrained window. The objects RIDGE, SLOPE_BREAK and GEOLOGIC_INTERFACE are searched for by querying the Spatial Database for Linear features. The UPPER_END_STREAM object is searched for by applying its special purpose search function. This function accesses FIRST_ORDER_STREAMS from the database and compares the elevations of their end-points.

If the spatial object has multiple components, as in the case of the query described above, one of two constraint satisfaction strategies is selected, currently under user control. In spatial constraint propagation mode, the set of sub-objects is ordered using the criteria discussed above. A backtracking search strategy modified to incorporate constraint propagation is used to search for sub-objects in the chosen order. Each object is searched for, using recursive calls to the high level search routine, in areas that are obtained by propagating constraints from the current locations bound to all earlier objects in the chosen order. In all cases the minimum bounding rectangle of the constrained search area is used as the rectangular query window and filtering using explicit constraint checking is used to remove extracted examples that do not satisfy the required spatial relationships. Backtracking search using constraint propagation continues till the required number of location-sets for the multi-component object have been found or all combinations of sub-object locations have been explored.

References

Abel, D.J. & Smith, J.L., 1983, A Data Structure and Algorithm based on a Linear Key for Rectangle Retreival, Computer Vision, Graphics and Image Processing, vol. 24, pp. 1-13.

Bentley, J.L., 1975, Multidimensional Binary Search Trees used for Assosiative Searching, Communications of the ACM, vol. 18, pp. 509-517.

Blackwell, B., 1987, The Use of Range-Tree Spatial Indexing to Speed GIS Data Retreival, AutoCarto 8, Baltimore, Maryland.

Brown, C.A. & Purdom, P.W., 1981, An Average Time Analysis of Backtracking, SIAM J. Comput., vol. 10, no. 3, pp. 583-593.

Chazelle, B. & Edelsbrunner, H., 1984, Optimal Solutions for a class of point retreival problems, Report No: CS-84-16, Dept. of Computer Science, Brown University.

Cole, R. & Yap. C.K., 1984, Geometric Retreival Problems, Information and Control, vol. 63, pp. 39-57.

Freuder, E.C., 1982, A sufficient condition for backtrack free search, Journal of the ACM, vol. 29, pp. 24-32.

Garey, M.R. & Johnson, D.S., 1979, Computers and Intractability - A Guide to the theory of NP-Completeness, W.H. Freeman and Co., New York.

Peng, G. & Smith, T.R., 1987, Space Efficient Hierarchical Structures : Relatively Addressed One to Four Autumnal Quadtrees for GIS, International Geographic Information Systems Symposium, Arlington, Virginia.

Haralick, R.M. & Elliott, G.L., 1980, Increasing Tree Search Efficiency for Constraint Satisfaction Problems, Artificial Intelligence, vol. 14, pp. 263-313.

Knuth, D., 1975, Estimating the efficiency of backtrack programs, Mathematics of Computation, vol. 29, pp. 121-136.

Mackworth, A.K., 1977, Consistency in Networks of Relations, Artificial Intelligence, vol. 8, pp. 99-118.

Mackworth, A.K. & Freuder, E.C., 1985, The Complexity of some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems, Artificial Intelligence, vol. 25, pp 65-74.

Matsuyama, T., Hao, L.V., & Nagao, M., 1984, A File Organization for Geographic Information Systems based on proximity, Computer Vision, Graphics and Image Processing, vol. 26, pp. 303-318.

McGregor, J.J., 1979, Relational Consistency Algorithms and Their Application in Finding Subgraph and Graph Isomorphisms, Information Sciences, vol. 19, pp. 229-250.

Menon, S. & Smith, T.R., 1987, Multi-Colored Quadtrees for GIS: Exploiting Bit-Parallelism for Rapid Boolean Overlay, International Geographic Information Systems Symposium, Arlington, Virginia.

Samet, H., 1984, The Quadtree and Related Hierarchical Data Structures, ACM Computing Surveys, vol. 16, pp. 187-260.

Smith, T.R., Peuquet, D.J., Menon, S. & Agarwal, P., 1987, KBGIS-II A Knowledge Based Geographic Information System, Internation Journal of Geographical Information Systems, Vol I, no 2, pp 149-172.

Waltz, D.L., 1975, Understanding line drawings of scenes with shadows, in The Psychology of Computer Vision, ed. Winston, P.H., pp. 19-91, McGraw-Hill.

Willard, D.E., 1978, Predicate oriented data base search algorithms, PhD. Thesis. Harvard University, Cambridge, MA.

Willard, D.E., 1982, Polygon Retreival, SIAM J. Comput., vol. 11, pp. 149-165.

BLANK PAGE

GLOBAL POSITION SYSTEMS

BLANK PAGE

GEOGRAPHICAL INFORMATION SYSTEMS AND NAD 83

Dr. John D. Bossler

Center for Mapping
The Ohio State University
Columbus, Ohio

ABSTRACT

The impact of the recently completed North American geodetic datum, NAD 83, on surveyors, mappers, photogrammetrists and users of geographical information systems (GIS) is significant. The major problem, of course, is the transformation of coordinate and cartographic information from the present datum, the North American Datum of 1927 (NAD 27) to the NAD 83. The optimum solution to the problem of transforming between the NAD 27 and the NAD 83 is to adjust, in a least squares sense, all available observational geodetic data either with a public domain software package or through the National Geodetic Survey and then use a computed transformation to shift the cartographic features from NAD 27 to NAD 83. The latter can be accomplished within or outside of the GIS.

1. BACKGROUND ON THE NAD83

After more than 350 staff years of effort, and over more than 10 years of time, North America has a new, "horizontal", geodetic datum known as the "NAD83". The best news related to this datum is:

1. All "available" geodetic control in the U.S. has been included. "Available" control is defined as monumented, described, of at least third order accuracy and submitted to the National Geodetic Survey (NGS) in digital form prior to mid 1981.

2. The results are much better than anyone expected; an average accuracy of 1 part in 300,000 between adjacent first order points (an inch in 5 miles or 1 cm. in 3 kilometers) has been obtained.

3. The datum is global, commensurate with the latest satellite technology and it spans Canada, Greenland, Mexico, and Central America. A recent, brief summary of the project is given by (Bossler, 1987) and other extensive documentation is under way.

There is also some "bad news" however. The surveying and mapping community along with a host of others involved in remote sensing, photogrammetry, automated mapping, cartography, users of Geographical Information Systems (GIS), etc., must deal with the fact that their spatial reference system, a system of geodetic (geographic) coordinates, has been changed. The shifts of latitude and longitude between the old reference system (NAD27) and the new system (NAD83) are shown in Figures 1 and 2. This presents a significant problem to the practitioner, which like most problems, should be faced as soon as possible.

EXPECTED LATITUDE CHANGE FROM NAD 27 TO NAD 83 (IN METERS)

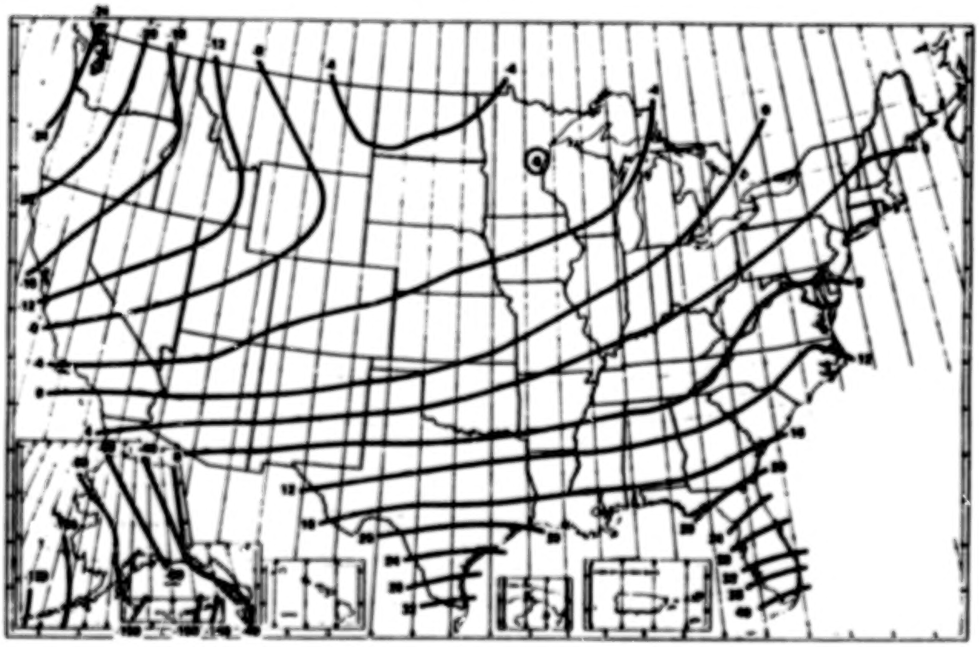
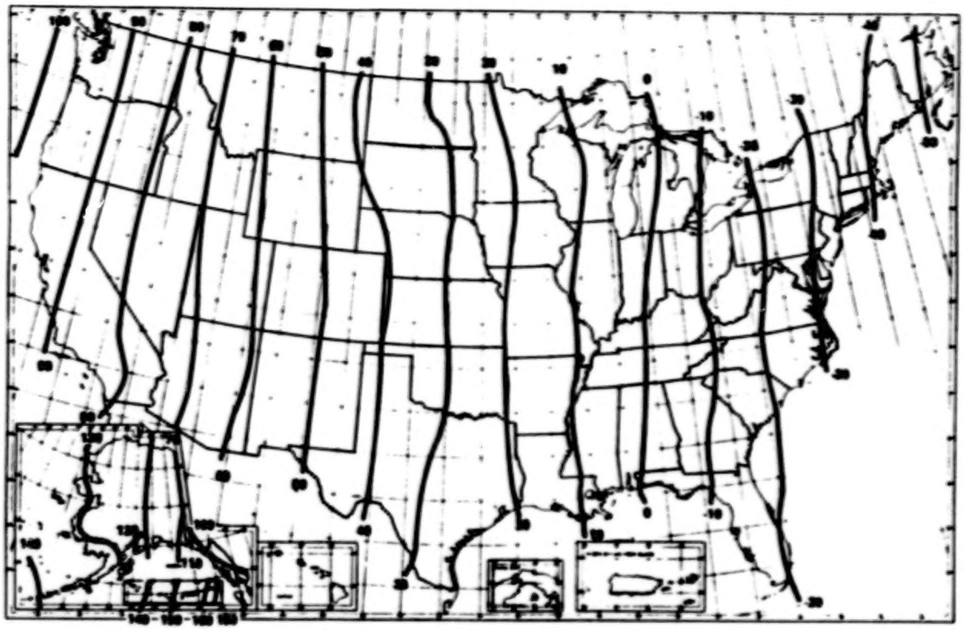


Figure 1

BEST COPY AVAILABLE



EXPECTED LONGITUDE CHANGE FROM NAD 27 TO NAD 83 (IN METERS)

Figure 2

2. DATA TYPES

There are two basic types of data that we need to consider:

- Traditional geodetic data consisting of distances, directions, azimuths and both astronomic and geodetic positions which were observed directly (e.g., Doppler satellite positions). We assume this data was not included in the NAD83 and we will call this "Type I" data.

- Positions which are scaled from a cartographic product or which have been estimated, etc. We will call this "Type II" data.

In principle, there is no difference between "Type I" positions of latitude and longitude and "Type II" positions. The difference is simply in the variance (uncertainty) (σ^2) of the position, i.e., $\sigma_A^2 << \sigma_B^2$.

BEST COPY
AVAILABLE

2.1 "Type I"

There are however, issues we need to consider between Type I and Type II data. Type I data can and should be rigorously adjusted into the NAD83. This is the recommended strategy by the NGS.

In order to accomplish this one needs to:

1. Key (digitize) the raw observational data in accordance with the Federal Geodetic Control Committee (FGCC) format commonly known as the Blue Book (Pfeifer, 1980).

2. Either submit this data to NGS who will generally adjust it "free of charge" or adjust the data with existing public domain software (Schwarz, 1978). If one does the latter, he is faced with potential conflicts between the NAD83 positions derived from the new data and the existing NAD83 positions. A number of ways of dealing with this problem are discussed in (Vincenty, 1987).

This author clearly recommends adjustment by NGS leaving the tricky problem of coordinate constraints, assignment of variances, etc. to the agency which is ultimately responsible for the network anyway. I assert that in general, the NAD83 coordinates will not change and the new data will be fitted into the NAD 83. The reason that the NAD83 coordinates will not change is because of their high accuracy (1/300,000) compared to the new data. This is true even for GPS surveys over 2-4 km where $\sigma_{GPS} \Rightarrow 1/100{,}000$. So much for Type I data. With rigorous new NAD83 coordinates available for all the control points, the task ahead is much simpler. At the end of the above process we will have NAD83 and NAD27 coordinates for a certain set of points of interest to us.

2.2 "Type II"

The problem of determining the NAD83 coordinates of cartographic features (houses, roads, etc.) given the NAD27 coordinates is equivalent to the general problem of transforming coordinates from one coordinate system to another. The usual problem of non-homogeneous systems applies in this case too! That is, the scaled or estimated cartographic data (Type II) are inconsistent. A number of methods are available to solve this problem starting with a similarity transformation (determining a rotation, two translations and a scale factor) or including more sophisticated techniques (Moffitt, 1987). It is important to note that NGS will assist practitioners with this effort also (Wade and Dole, 1987). However, it is at this point that we can utilize most GIS's themselves.

3. USING THE GIS TO TRANSFORM SYSTEMS

If we have Type I data and we determine the NAD83 coordinates for our points using the procedures outlined above, and they are reasonably geographically distributed over our area of interest, the problem is straight forward. To the best of my knowledge most GIS's are capable of solving this problem. For illustrative purposes and because I am familiar with it, I will discuss the Synercom system known as ZORRO. It uses an Affine transformation of between four (θ, x, y, k) and six (θ, x, y, k_x, k_y, γ) parameters. Here

θ = Rotation angle between coordinate systems

x = Translation in the x coordinate

y = Translation in the y coordinate

k, k_x, k_y = Scale changes (globally, in x only, in y only)

γ = The (sometimes other than 90°) angle of the "new" coordinate axis.

For most applications over reasonably small areas $k = k_x = k_y = 1$ and $\gamma = 90°$. We provide our data in terms of (usually) two sets of plane coordinates: $\vec{X}_{NAD27}$, $\vec{X}_{NAD83}$ and obtain T = θ, x, y, and perhaps k, etc., from ZORRO. Here the form of the equations is:

$$\begin{bmatrix} X_{83} \\ Y_{83} \end{bmatrix} = k \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_{27} \\ Y_{27} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

and

$$T = \begin{bmatrix} x \\ y \\ \theta \\ k \end{bmatrix}$$

ZORRO is of course, a least squares solution. We can now use our transformation T to convert all NAD27 Type II coordinates to NAD83 coordinates in the same area. In this process we assume that T is "perfect", obtained from two sets of geodetic quality (Type I) data.

Incidently, ZORRO is usually used to transform digitizer coordinates to either NAD27 or NAD83 coordinate systems.

I have glossed over several points. First, if the area of interest is large we may wish to divide it into homogeneous smaller regions so that our transformation is valid and representative of the data set. Here "validity" can be determined by ZORRO also, from an inspection of the residuals from the transformation parameter solution. These will show how "linear" or at least how smooth the coordinate differences tended to be. In general, coordinate differences over areas of about 50 km by 50 km (30 mile by 30 mile) should be representable by one transformation.

Another problem exists if the coordinates in both systems are not available over the entire area of interest. This of course is a serious problem which can be dealt with rigorously, i.e., add geodetic control in the area or dealt with superficially, i.e., assume one transformation holds over the entire area.

The last situation that we need to consider is if we have only NAD27 coordinates which are Type II. We will need to to obtain some NAD83 coordinates of the same points. A reasonable minimum of 4 points is needed; however in general, the more common points, the better determined the transformation will be (ZORRO has an upper limit of 30). Therefore scaling or estimating the NAD83 coordinates of 4 or more points in each area of interest is required. ZORRO can be used to determine the transformation T, for an area of interest. It is not clear from the documentation on ZORRO whether or not ZORRO can be utilized with weights $(1/\sigma^2)$ for each coordinate set (NAD27 and/or NAD83). However correspondence with Charles Drinnan of Synercom suggests that this is possible. If possible, one would input $\vec{X}_{NAD83}$, Σ_{NAD83} and $\vec{X}_{NAD27}$, Σ_{NAD27} and determine T. The input of the variance data may well be important in this case as either the NAD83 coordinates or NAD27 coordinates could be much more accurately determined than the other. If the program is completely general each coordinate pair could be assigned a different weight. Here relative weights are to be used - not those referred to the origin of the global coordinate system. In this context relative refers to a local coordinate system.

4. SUMMARY

The new adjustment and redefinition of the North American horizontal geodetic network presents a challenge to those working with coordinates. Most GIS are capable of assisting practioners by providing them with built-in transformation software, hence we have moved one step more toward centralizing and focusing our overall problem. We look forward to further advances in GIS's so that additional problems associated with cartographic feature placement, etc. can be facilitated and then finally automated. A series of cases is summarized in Table 1.

Case	Available Information	Recommended Procedure
1	$\vec{X}_{NAD83}$, Type I and Type II data. Reasonable geographical coverage of stations. Reasonably small area. (< 50 km x 50 km)	1. Have NGS compute the NAD83 coordinates using the Type I data and incorporate this data into the national network. 2. Let GIS compute transformation T, and use T to transform all Type II data.
1a	Same as Case 1 but very large area (> 50 km x 50 km)	Same as Case 1 but subdivide area into smaller units and determine $T_{i=1,n}$ where n = the number of units.
1b	Same as Case 1 but poor distribution of control points.	1. Add new control points in sparsely controlled areas. 2. Follow Case 1.
2	$\vec{X}_{NAD27}$[1], Σ_{NAD27}[2] Type II data only. $\Sigma_{NAD27} \neq \Sigma_{NAD83}$	1. Scale or estimate $\vec{X}_{NAD83}$ for > 3 common points. Also determine Σ_{NAD83} from scaling procedure. 2. Use GIS to compute T using $\vec{X}_{NAD27}$, Σ_{NAD27} and $\vec{X}_{NAD83}$, Σ_{NAD83} 3. Transform remainder of Type II data using T.

[1] If $\vec{X}_{NAD83}$ has been obtained, case 2, step 1 can be omitted.

[2] Obviously Σ_{NAD27} must be estimated from scaling procedure.

Table 1.

References

1. *Bossler, J.D.*, "Geodesy Solves 900,000 Equations Simultaneously," EOS Transactions, American Geophysical Union, Vol. 68, No. 23, June 9, 1987.

2. *Moffitt, F. and S. Jordan*, "Conversion of State Plane Coordinates from NAD27 to NAD83 by Second Degree Conformal Transformation," ACSM Bulletin No. 106, p. 13-20, 1987.

3. *Pfeifer, L.*, "Input Formats and Specifications of the National Geodetic Survey Data Base," Vol. I, Horizontal Control Data, Federal Geodetic Control Committee, National Geodetic Information Center, NOAA, Rockville MD., 1980.

4. *Schwarz, C.*, "The TRAV-10 Horizontal Network Adjustment Program," National Geodetic Survey, NOAA, NOS 73, NGS 8, Rockville Md., May, 1978.

5. *Vincenty, T.*, "Conformal Transformations between Dissimilar Plane Coordinate Systems," submitted to the Journal of Surveying and Mapping of the American Congress on Surveying and Mapping, 1987.

6. *Wade, L. and D. Doyle*, "Datum Transformations from NAD27 to NAD83," Technical Papers of American Congress of Surveying and Mapping, Vol. 3, p. 27-36,, 1987.

GIS POINT REFERENCING BY SATELLITE AND GRAVITY

Alfred Leick
University of Maine
120 Boardman Hall
Orono, ME 04469

ABSTRACT

A brief review of the Global Positioning System (GPS) is given, including its current and expected capability and the research required to make optimal use of GPS satellite observations, such as the pseudo ranges and carrier phases for GIS purposes. This includes a discussion of static and kinematic techniques of GPS surveying. Since GPS does not provide (orthometric) heights, there is a need to use gravity observations to supplement GPS observations. GPS and gravity observations are supplementary and constitute a unified measurement system for global 3-dimensional point mapping. GPS capability leads to a rethinking of the use and purpose, as well as the form and shape of conventional geodetic networks, e.g. super network or no network at all. Respective research topics are identified.

1. INTRODUCTION

Absolute or relative positioning is at the base of a Geographic Information System (GIS). The requirements range from coarse to highly accurate positioning across the spectrum of GIS applications, and sometimes even within the same application. The Navigation Satellite Timing and Ranging (NAVSTAR) Global Positioning System (GPS), which is under development by the Department of Defense, will be the major tool for positioning points world-wide on a 24 hour per day basis, under all weather conditions, on land, in the air, and on the sea. In spite of its primary military mission, GPS will have profound implications upon the everyday lives of civilians. Despite the current suboptimal GPS satellite configuration, the GPS system has already astounded even its staunchest supporters.

2. THE NAVSTAR GPS SPACE SYSTEM

At the present time 10 Block I GPS satellites are in orbit in order to develop and test user equipment and algorithms. Several of these "experimental" satellites are operating well and far beyond their design live time. Two of these satellites are not usable any more, and one is only of marginal use. The Challenger accident postponed full deployment of the Block II GPS satellites by about 2 years. The latest launch schedule (8) calls for the first Block II satellite to be launched in the last quarter of 1988. Full deployment of all satellites is now expected by the end of 1991. Launch vehicles will be the Delta II rocket and the space shuttle.

The original plans called for a 24 spacecraft constellation. Budgetary considerations resulted in a reduction of the number of space vehicles to 18 plus 3 active spares. Calculations show that the latter constellation, despite all geometric optimization, would have, at any given time, at least 4 areas of degraded navigation performance around the globe. These so-called "outage" areas would change in size, duration and location as a function of time, but would be accurately predictable. Other navigation tools can be used to bridge these outage areas. The military is reexamining their requirements closely as to the need to revert back to the original 24 satellite constellation (7). No doubt, this is also an opportunity for the civilian aviation community to contribute to the navigation system they will be using tomorrow. Other criteria such as survivability, ease of clock synchronization, and better isolation of failures also point at a 24 satellite configuration as the most desirable one.

The satellites will be in a circular orbit with an altitude of about 20.000 km resulting in an orbital period of about 12 hours. The orbital inclination will be 55 degrees; there will be 6 equally spaced orbital planes. The distribution of the satellites within the orbital planes will be such that the 24 hour operational capability will be assured. The satellites transmit coherently at frequencies L1=1575.42 MHZ and L2=1227.6 MHZ derived from a fundamental frequency of 10.23 MHZ and modulated with several signals. The L1 and L2 carrier are modulated with a pseudo-random noise (PRN) code, called the P-code. Its primary use is for high precision and instantaneous navigation. This code is highly resistant to jamming. The L1 carrier is additionally modulated with the coarse/acquisition C/A-code. Both carriers contain the navigation message which contains information on the health of the satellites, the latest up-date information, time corrections, the position of the satellite as well as the positions of all the other GPS satellites. Most receivers can read the navigation message. Details on the signal structure can be found in reference (22).

3. GPS RECEIVER EQUIPMENT

New receivers are introduced at almost every major GPS meeting; some of the existing receivers are already outdated and slated for retirement. Some receivers require knowledge of the P-code, others use only the C/A-code, and there are codeless receivers requiring no knowledge of either code. Depending on the features a particular receiver exhibits and the software provided, the receiver may be useful for navigation, surveying, for time transfer, or for all three applications.

There are currently 25 manufacturers of GPS receivers offering a total of 58 different receivers (16). Table 1 has been compiled using the same reference. It lists the application (navigation, survey, time) versus the use environment (land, marine, aeronautical). This table can be interpreted as the manufacturers view of the needs and uses of GPS (following expectations of sale and market analysis). It is seen that the application "navigation" and the environment "land" are most often considered. The time transfer capabilities with GPS seem to weigh heavily in the minds of the manufacturers.

Table 1: Application / Use Environment Distribution of Receivers

	L	M	A
N	32	29	22
S	23	11	7
T	23	14	11

N=Navigation S=Survey T=Time
L=Land M=Marine A=Aeronautical

Table 1 contains a subset of about 12 receivers capable of measuring carrier phases, which is a requirement for high precision geodetic applications.(This includes receivers which are no longer manufactured, and receivers which become available within 1987). This number becomes even smaller if one counts only those receivers which are able to measure carrier phases on L1 and L2 (a requirement to eliminate the ionospheric effect). Thus, one should expect that receivers for high precision surveying remain expensive (at least more expensive than the typical navigation receiver) for the near future unless the market analysis changes.

The size, weight and power consumption of the receivers is steadily decreasing. Table 2 gives some approximate numbers. Roughly speaking, one can say that the first generation of GPS receivers weighed about 120 pounds, the second generation brings about 30 pounds on the scale, and the 1987 class of receivers is less than 10 pounds. The current record is held by a 3-pound, 3-watts receiver. The trend toward miniaturization is continuing.

Table 2: Miniaturization of Receivers

Receiver Generation	Weight (pounds)	Power (watts)
first	120	215
current	30 - 40	25
1987	10	10
	3	3

4. PSEUDO RANGES AND CARRIER PHASE OBSERVATIONS

The pseudo range observable equals the distance between the satellite and the receiver, plus a small corrective term due to receiver clock offset. The pseudo ranges are derived from the travel time of the P-codes or the C/A-codes. Given the geometric positions of the satellites at the time of signal transmission, four pseudo ranges are sufficient to compute the three position coordinates of the receiver and the receiver clock offset. This technique is primarily used in instantaneous navigation (in which case the velocity of the vehicle is also derived from the same signals). As shown in Figure 1 one receiver is sufficient for this mode of operation.

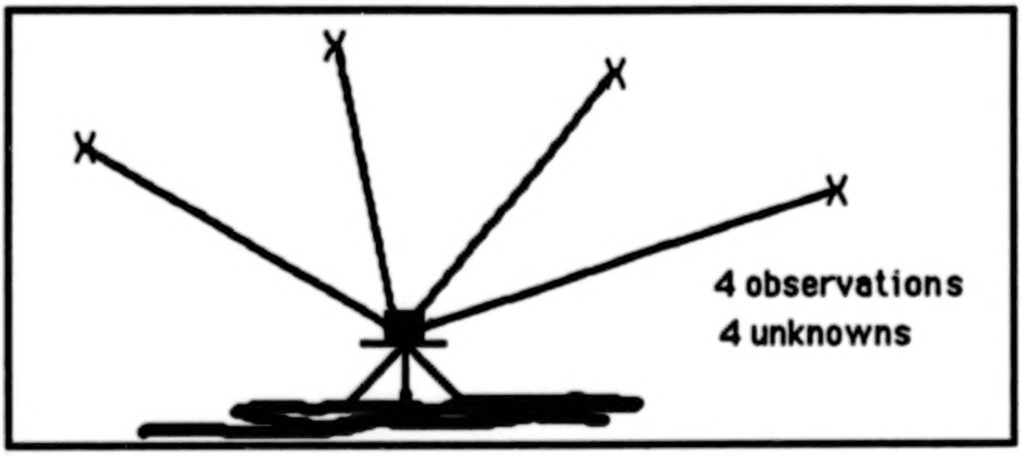


Figure 1: Navigation Solution With GPS

The carrier phase observable is the difference between the phase of the carrier signal of the satellite and the phase of the local oscillator within the receiver at the epoch of measurement.

Receivers are programmed to make phase observations at the same and equally spaced epochs. In addition, receivers keep track of the number of complete cycles received since the beginning of measurement. Thus, the actual output is the accumulated phase observable at pre-set epochs. This quantity equals the integrated doppler of the carrier plus the fractional phase. The measured quantity contains occasional jumps equal to an integer number of cycles. This can be caused by temporary blockage of the signal or by weak signals. If this happens, a cycle slip is said to have occurred. There are postprocessing techniques which effectively deal with cycle slips.

Much is gained by forming linear combinations of the basic phase observable, e.g. unwanted parameters are eliminated and certain effects need not be modelled. Assume that the receivers l and m have observed the same satellite p at the same time. The difference of these two phase observations is called a single difference. It can readily be shown that single difference observations are largely independent of satellite frequency offset and linear drift. Assume that one has two single differences referring to satellites p and q respectively. The difference of these two single differences, called the double difference observation, is largely independent of receiver clock errors. Finally, taking the difference of two double differences referring to different epochs, yields the triple difference observations. A typical arrangement is shown in Figure 2. At least two receivers observe simultaneously. There is no need for visibility between the observing stations. Only receivers which observe at least the phases of the L1 carrier are able to provide centimeter level baseline accuracy.

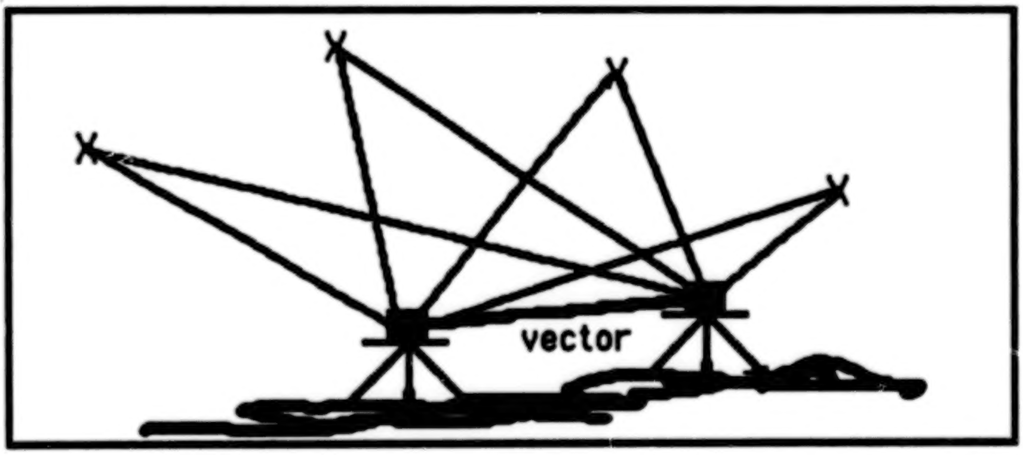


Figure 2: Relative Positioning With GPS

Government announcements have been made regarding Selective Availability (SA). This involves deliberate variation of the satellite transmissions, essentially by dithering with the clock effecting both the P-code and the C/A-code. Code receivers might require supplemental cryptographic technology in order to perform to their fullest potential in real-time. It is the understanding of this author that during January and February 1988, SA will be turned on providing an opportunity to study the impact of SA on civilian receivers and algorithms. More detail on the civil access to GPS codes is found in (4).

5. GPS SURVEYING TO DATE

The secrets of GPS success can be readily explained. At the base is the ability to measure carrier phases to about 1/100 of a cycle which equals 2 to 3 millimeters in linear distance. The high frequencies L1 and L2 penetrate the ionosphere relatively well. Since the impact of the ionosphere is a function of the frequency, observations at both frequencies can be used to eliminate most of the ionospheric effect. Another reason for the success of GPS is the progress in the design and miniaturization of clocks in recent years, providing a very stable reference for the receivers. Also, GPS satellite orbits are stable due to their high altitudes and can be computed (predicted) fairly accurately. There are no atmospheric drag effects, and only the lower spherical harmonics of the gravity field expansion contribute to the gravitational attraction. The impact of the sun and the moon on the GPS orbits is significant, but can be computed accurately. Thus, the (only) remaining worrisome effects are those of solar radiation pressure and troposphere.

Table 3 summarizes GPS surveying capability. The outcome of a typical GPS navigation solution are geocentric positions accurate to a few meters. The geocentric positions refer generally to the WGS84 or WGS72 datum. The attainable accuracy depends on whether the P-code or the C/A-code is used for pseudo ranging, and on the station-satellite geometry (Other factors are the length of observations for a stationary user, and whether dual band observations are possible). It is important to note that under the conditions of Selective Availablity, the attainable accuracy is expected to degrade to about 100 m.

Table 3: GPS Positioning Capability

Navigation Solution
5 - 16 m 100 m (Selective Availability)
Static Relative Positioning
1 ppm (< 15 minutes) .1 ppm by research groups .01 ppm goal?
Kinematic Relative Positioning
1 m carrier aided code observations 1-2 cm receiver on the ground 5-10 cm receiver in the airplane

The primary result of single, double, or triple difference carrier phase processing is the vector between the observing receivers. The accuracy is generally expressed by a constant part, which is largely dependent on the antenna design, and a part which is proportional to the distance between

BEST COPY AVAILABLE

the receivers. Currently 1 part per million (ppm) is achieved routinely by practitioners, whereas various research groups achieve .1ppm. The limit of GPS relative positioning accuracy is expected to be about .01ppm.

In 1983 GPS Surveying at the 1-2ppm accuracy level was verified beyond any doubt. A 35 station first-order geodetic network densification in the Eifel region of West Germany was observed (1). This project was a joint effort by scientists and a private U.S. firm. In early 1984 a 42 station network densification of Montgomery County (Pennsylvania) was carried out. The sole guidance of this project rested with a private GPS surveying firm (3). In 1984 GPS was also used at Stanford University for a high precision GPS engineering survey to support construction extending the Stanford Linear Accelerator (SLAC). Terrestrial observations (angles and distances) were combined with GPS vectors in one solution. The Stanford project yielded the first truly millimeter accurate GPS network, demonstrating the high quality of the antenna used. This accuracy is confirmed through comparison with the alignment laser of the accelerator which reproduces a straight line within one tenth of a millimeter (21). A network survey using dual frequency receivers is reported in (9,10) claiming 1 ppm in all three components of a vector in as little as 15 minutes of observation time. The accuracy level of .1 ppm has been achieved by several research groups. A repeatability of 2-5 part in 10^8 in all three components is reported in (2,15). Lichten and Border (15) give the specific example of a 246 km baseline which was measured repeatedly to 0.4-0.6 centimeters. Another example was an 1509 km baseline measured repeatedly to 2-3 centimeters. Solutions of this kind are presently only available to researchers. However, considering the rapid advancement in GPS implementation, one can realistically expect these kind of solutions to be available to the practitioners in about 2 to 3 years. Such highly accurate solutions require the estimation of GPS orbital improvement parameters simultaneously with baselines and careful consideration of the effect of the troposphere and the ionosphere.

One accurate vector by itself is generally not of much use. Of course, one can add the vector to the geocentric position of the "known" station and formally compute the geocentric position of the new station. The problem with this procedure is that the uncertainty of the "known" station is transferred in full to the new station. Also, the vector observations themselves can be in error despite all of modern technology. Possibilities of misidentification of stations, centering errors, misreading of antenna heights, etc can never be completely disregarded. Like any other observations, the GPS vector observations are most effectively used and controlled in a network of stations whose relative positions are determined by vector observations only and which are subjected to a minimal or inner constraint adjustment (13). Such an adjustment makes it possible to assess the quality of the observations, validate the correctness of the covariance matrices and detect and possibly remove blunders. The inner constraint solution has the added advantage that the resulting covariance matrix of the adjusted coordinates is independent of the coordinate system definition and truely reflects the inner geometry of the network and the geometry of the satellite configuration. Thus, the primary result of a GPS survey is a polyhedron of stations whose relative locations are accurately determined.

Accurate positioning of moving platforms is the newest application of GPS. Kinematic differential GPS surveying is a prime example. This technique uses carrier phase observations to determine the position of a roving antenna with respect to a stationary antenna. Both receivers record the carrier phase observations continuously. The only constraint on the path of the moving antenna is visibility of the satellites. First experiments in kinematic differential GPS yielded centimeter relative accuracy in seconds (20). Initial relative antenna positions are obtained in seconds through the antenna-swapping technique. The theory and software of kinematic differential GPS has been refined at the University of Maine (18). Kinematic differential GPS has been used for decimeter positioning of photogrammetric airplanes (6,17). High - precision airplane positions eliminate or reduce the need for conventional ground control in photogrammetry.

The kinematic differential techniques referred to above are very accurate but are sensitive to cycle slips in the carrier phase observable and require a minimum of satellites visible at any time. An alternative is to use carrier observations to aid (supplement) code observations (19). Carrier phase observations to the same satellite and pseudo range differences describe the same station - satellite range differences in time. Code observations are less accurate and can suffer from multipath problems, whereas carrier phases can be measured accurately, are little effected by multipath, but may contain cycle slips. Thus, both types of observations are supplementary and can result in a robust measurement system yielding about one meter positional accuracy.

Much research is carried out combining GPS receivers with inertial units. Most of the work deals with the combination of code observations and the inertial unit. Perhaps inertial units can be effectively used in the near future to correct cycle slips in the carrier phase observations. This would then result in a very robust but accurate system. Other hybrid arrangements combining GPS receivers with an Omega or Loran C navigation set are already available.

6. THE GRAVITY COMPONENT

As a by-product, GPS might soon make the use of leveling to the 3-4 centimeter accuracy obsolete. As explained above, the primary result of a GPS survey is a polyhedron of stations with precisely known relative locations as shown in Figure 3. These relative geometric locations can be referenced to an ellipsoid in terms of ellipsoidal coordinates, i.e. latitude, longitude, and height. As is well known, the ellipsoidal height differs from the orthometric height by the separation of ellipsoid and geoid, called geoid undulation. Thus, it would be desirable to compute geoid undulations as accurately as ellipsoidal heights can be derived from GPS measurements. Unfortunately, this is not yet possible.

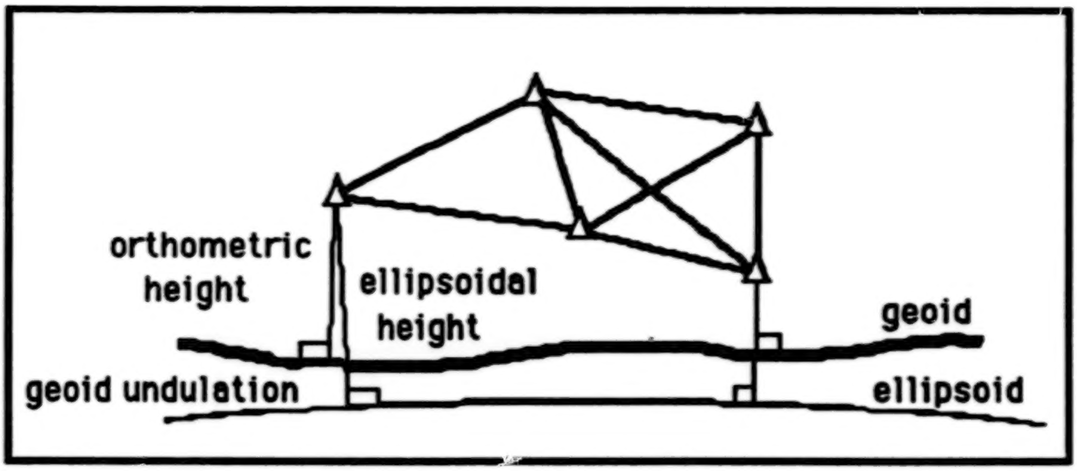


Figure 3: Ellipsoidal Heights vs Orthometric Heights

The geoid is an equipotential surface of the sum of the gravitational potential of the earth and the centrifugal potential. Thus, the undulating features of the geoid are a direct result of the over-all mass distribution within the earth. Determining the shape of the geoid, i.e. giving a mathematical expression for the geoid undulations is part of ongoing research in geodesy. Currently, several spherical harmonic solutions up to degree and order 180 are available, with research being conducted on solutions of degree and order 360. The higher the degree of the spherical harmonic solution the more geoidal features are represented. These so-called global solutions are based on mean (average over small area) gravity values and satellite observations (primarily satellite altimeter data). Techniques have been developed to supplement these global solutions with local gravity observations to compute high precision local geoid undulation differences (5, 11, 12). Such solutions can reach the 2-3 centimeter accuracy level.

7. GEODETIC RESEARCH ISSUES AS RELATED TO GIS

7.1 Network Maintenance/Densification

Incorporating new observations into an existing network is an old problem which has been around ever since K. F. Gauss adjusted the geodetic network of Hannover (Germany) centuries ago. Many statistical procedures have been suggested and used to make "distortions disappear". All too often the usefulness of "good measurements" is recognized too late for these multi-user multi-purpose geodetic networks. In general, the GPS network accuracy of 1-2 ppm exceeds that of the existing first order geodetic network. The goal of network maintenance should be to use GPS observations to gradually obtain a better network rather then to degrade high-quality GPS observations to fit the existing network of lesser quality.

Implementation at the coordinate level is most common among surveyors. The new satellite - derived positions and the existing geodetic network positions are related through appropriate transformations and scaling using common stations, i.e. stations whose positions are known in both coordinate systems. The more common stations there are available, the better the transformation is defined. Implementation at the level of the observations through readjustment of old and new observations is certainly practical for a local region of the size of a county.

Research is required to make network adjustment of local regions available to those who generate the measurements. Since the correct treatment of observations requires a deep but definable knowledge of adjustment theory and geodetic concepts, there is welcome opportunity here to experiment and eventually to use fashionable tools such as expert systems and/or artificial intelligence. There is a need to identify the best database designs and computer supported telecommunication requirements for the exchange of measurements among surveyors and, possibly, a national or county-wide database (14). Databases must have optimal capability for making available all observations and related data within neighborhoods defined by the user. The challenge is to create a correct yet transparent and user-friendly network maintenance capability.

7.2 Automated GPS Base Stations

To the creative mind, the expected GPS capability of 1 part in 10^8, i.e. 1 millimeter in 100 kilometers, opens the door to new forms of geodetic networks. One often hears the following:" Put a receiver on the roof of the court house and let it record measurements continuously. All boundaries will be referenced right to the court house." The fact that data collection at one end would be controlled by "authorized personnel" could possibly be used advantageously in formulating objective quality control and gaining legal acceptance or a higher legal weight of measurements. Other people speak of super networks covering the whole nation and provide super-stations as references. These kinds of unconventional approaches have the distinct advantage that the user must invest at a minimum in one receiver. The super-stations would be equipped with dual frequency receivers, operate autonomously, and would make the observations available in a database directly accessible to the user.

The super-station idea could conceivably eliminate the need to occupy existing geodetic monuments. Locating and verifying old network stations is a major expense in any GPS survey. This is particularly true in remote areas which have not experienced many geodetic activities in the past. The user could rent a dual frequency receiver to establish the precise connection to the next super-station, and then use his own single frequency receiver(s) to complete the local survey.

Automated GPS base stations would also be useful in computing the time and geographical variations of the effect of the ionosphere on single frequency GPS user. Research is needed on the feasibility and use of the automated super-station concept. Pure economic considerations might, indeed, very soon force surveying in this direction.

7.3 A More Accurate Geoid

The use of local gravity to compute high precision geoid undulation differences to supplement GPS derived ellipsoidal heights still needs to be investigated thoroughly. First results are very encouraging. There is an obvious need to look, once again, at the quality and distribution of available gravity data. Gaps in the gravity data will hopefully be filled in the near future using gradiometers in airplanes and other techniques.

7.4 Real-Time GPS Surveying

GPS was conceived by the military as a real-time (instantaneous) positioning system with use of the P-code yielding positional accuracy in the meter range. Because the P-code will not be made fully available to the civilian community for reasons of national security, alternatives for real-time GPS surveying must be developed. The L1 C/A-code receivers which are equipped with or supplemented with a communication link to provide real-time differential accuracies of 1 meter, are a step in this direction. Real-time centimeter or even millimeter GPS surveying techniques must be based purely on carrier phase measurements. Industrial applications of real-time high - precision GPS surveying are plentiful. Examples are navigating construction or farming equipment, guidance of robots, and deformation monitoring.

8. SUMMARY

Since "high tech" is at the heart of GPS surveying, the user hardware will eventually be low cost and affordable by every professional. Combined with desktop computing power, electronic communication between databases, and prudent use of expert systems and/or artificial intelligence, GPS will open new frontiers in positioning. The possible uses of instant and accurate positioning combined with objective and transparent quality control stretches the imagination. Final judgement on GPS must wait until it is fully operational and users have had an opportunity to unleash innovative applications. Because of its over-riding military significance, full development is likely to be completed as planned. Because of the extreme accuracy needs in geophysical and other scientific applications, the high-precision spectrum of GPS capability will enjoy continued support and development. Real-time GPS surveying might even receive the most attention because of its close connection to navigation.

The user will have the choice of using instantaneous positioning (point positioning) of 100 meter accuracy down to instantaneous relative positioning of 1 meter of moving platforms using differential techniques based on C/A-code observations with possible carrier aided smoothing. Static relative positioning techniques are currently at the 1ppm level and require only 15 minutes of observation. Static techniques are expected to approach the .01ppm level. High precision differential techniques relying soley on the carrier phase observations will make instantaneous centimeter accurate relative positioning available to many applications on the ground and in the air.

References

1.Bock, Y., Abbot R.I., Counselman C.C., Gourevitch S.A., King R.W., 1985. "Establishment of Three-Dimensional Geodetic Control by Interferometry with the Global Positioning System", Journal of Geophysical Research, Vol. 90, No. B9, pp. 7689-7703

2.Beutler, G., Gurtner, W., Rothacher, M., Schildknecht, I.B., 1986. "Evaluation of the March 1985 High Precision Baseline HPBL) Test: Fiducial Point Concept versus Free Network Solutions". EOS Transactions, American Geophysical Union, 67,, Nov. 4, p. 911

3.Collins J., Leick A., 1985. "Analysis of Macrometer Networks with Emphasis on the Montgomery (PA) County Survey". Proceedings: First International Symposium on Precise Positioning with the Global Positioning System, Positioning with GPS - 1985, Washington, D.C., April 15-19

4.Ellett M. J., 1986. "Civil Access to the Precise Positioning Service of the NAVSTAR Global Positioning System". Proceedings: Position, Location and Navigation Symposium, Las Vegas, November 4-7.

5.Engelis T., Rapp R. Bock Y., 1985. "Measuring Orthometric Height Differences with GPS and Gravity Data". Manuscripta Geodetica, pp. 187-194

6.Hein G., 1987. Private Communication; University of the Federal Armed Forces, Neubiberg, W. Germany

7.Jensen, O.E., 1987. "GPS Overview - The Operator's Perspective". Presentation at the Institute of Navigation (Satellites Division) Technical Meeting on GPS Navigation, Colorado Springs, Sept. 23-25.

8.Jewkes S.D., 1987. "Countdown To Launch of GPS-13". Presentation at the Institute of Navigation (Satellites Division) Technical Meeting on GPS Navigation, Colorado Springs, Sept. 23-25.

9.Ladd, J.W. and Counselman C.C., 1985. "The Macrometer II Dual-Band Interferometric Surveyor". Proceedings: First International Symposium on Precise Positioning with the Global Positioning System: Positioning with GPS - 1985, April 15-19, 10.Ladd J.W., 1986. "Establishment of a 3-dimensional geodetic network Using the Macrometer II Dual-Band Surveyor". Bulletin Geodesique, Vol. 60, No. 3

11.Lambert S., 1987. Monitoring Vertical Crustal Deformation in Eastern Maine Using GPS Derived Orthometric Heights. University of Maine, Dept. of Civil Engineering, Report No. 71

12.Landauer H., Hehl K., Eisfeller B., Hein G., 1987. OPERA 2.3. User's Guide. University of the Federal Armed Forces, Neubiberg, W. Germany

13.Leick A., 1984. "Macrometer Satellite Surveying", Journal of Surveying Engineering, ASCE, Vol. 110, No.2

14.Leick A., 1984. "GPS Surveying and Data Management". Proceedings: URISA'84, Conference of the Urban and Regional Information Systems Association, Seattle

15.Lichten S.M., Border J.S., 1987. "Strategies for High Precision GPS Orbit Determination". Paper presented at the American Geophysical Union spring meeting, Baltimore, May 18-21

16.McDonald K.D., Parkinson B.W., McDonald C.P., 1987. "A Survey of GPS User Equipment, Applications and Receiver Technology Trends". Presentation at the Institute of Navigation (Satellites Division) Technical Meeting on GPS Navigation, Colorado Springs, Sept. 23-25.

17.Mader G.L., 1986. "Dynamic Positioning using GPS Carrier Phase Measurements". Manuscripta Geodetica, Vol. 11, No. 2

18.Quirion C., 1987. "Centimeter-Level GPS Surveys in Seconds: Formulation and Analysis". Proceedings: ASPRS/ACSM Annual Meeting, Baltimore, March 29 - April 3, 1987

19.Pietraszewski D., Spalding J., Viehweg, Luft L., 1987. "Status of United States Coast Guard Sponsored Differential GPS Demonstration System Development". Presentation at the Institute of Navigation (Satellites Division) Technical Meeting on GPS Navigation, Colorado Springs, Sept. 23-25.

20.Remondi B.W.,1985. "Performing Centimeter-Level Surveys in Seconds With GPS Carrier Phases: Initial Results". NOAA Technical Memorandum NOS NGS-43

21.Ruland R., Leick A., 1985. "Application of GPS to a High Precision Engineering Survey Network". Proceedings: First International Symposium on Precise Positioning with the Global Positioning System, Positioning with GPS - 1985, Washington, D.C., April 15-19

22.Spilker, J. J., 1978. "GPS Signal Structure and Performance Characteristics", Navigation, Vol. 25

BLANK PAGE

REMOTE SENSING

BLANK PAGE

ELECTRONIC BROWSING FOR SUITABLE GIS DATA

Jeffrey L. Star, David M. Stoms, Mark A. Friedl, and John E. Estes

Remote Sensing Research Unit
Department of Geography
University of California
Santa Barbara, CA 93106 USA
(805) 961-3845

ABSTRACT

With the mass of remotely sensed data being collected, researchers are often unaware of what data is available and where it is stored. Data is frequently purchased on the basis of a terse database record describing, for instance, cloud cover and image quality. A better approach may be to allow analysts to preview spatial data, not just some salient facts about that data.

In a NASA-sponsored research program, the authors are developing a prototype system with browsing capability for data archives. The objective is to allow scientists, sitting at their local workstations, to access a network, to retrieve records of image and spatial data selected by user-specified attributes, to view low resolution versions of the data, and to place an order, if the data are satisfactory. The architecture and functioning of the BROWSE testbed is briefly outlined.

Surprisingly, the definition of BROWSE has found no consensus. This paper focuses on the on-going efforts to establish a definition that satisfies a broad range of scientific disciplines. Any operational definition will include the algorithms used to compress the raw data into a low resolution format. Preliminary algorithms being considered include single band subsetting, spatial subsampling, band ratios, principal components analysis, and linear combinations. Several browsing scenarios illustrate the complexities of selecting suitable data sets. An effective browsing utility will have benefits beyond NASA data systems and the Earth Observing System. Lessons learned from this project may be of value to other spatial data base designers.

1. INTRODUCTION

Effective information systems are an increasingly important aspect of spatial data analysis (Estes, 1985). Technical and social innovations provide a forcing function for improvements in the design of information systems, driven by the increasing demand for and generation of scientific data. Naisbitt (1982) discussed several social transformations that characterize corresponding changes in how geographical analysis is being done. Among these trends are shifts from:

- centralization to decentralization
- hierarchies to networking
- an industrial to an information society and
- forced technology to high technology and 'high touch', meaning increased personal involvement.

The National Aeronautics and Space Administration (NASA) has recognized these trends in establishing pilot data systems for ocean, climate, land, and planetary science data (Jet Propulsion Laboratory, 1986a; NASA Goddard, 1986; NASA, 1986; Jet Propulsion Laboratory, 1986b). Working groups are also designing information systems for the vast quantity of data projected from the Earth Observing System (EOS) on the Space Station complex in the coming decade. The challenge is twofold. The first challenge lies in developing data handling techniques for the anticipated large data volumes. The second challenge is to bring this new technology to a broader scientific and applications constituency in the service of global science (Estes et al, 1986).

The Committee on Data Management and Computation (CODMAC) of the National Academy of Science predicts that an important capability in future spatial information systems will be the ability to browse through databases from a remote terminal (CODMAC, 1982). Browsing could perform three important functions. It would allow users to locate and preview spatial data and make an informed decision about their utility, for instance, as input to a geographic information system (GIS). Secondly, it would provide a mechanism for the user community to view newly received data and make recommendations on whether the data are suitable for permanent archiving. A third function of interactive browsing would be to aid decisions regarding what additional observations to acquire during an ongoing mission (CODMAC, 1986). At NASA, this mode of operation is being called *telescience*. However, different disciplines do not necessarily share common ideas on optimal browsing formats and characteristics. This paper describes attempts to define 'browsing' in the context of multidisciplinary spatial data systems. This research is part of an on-going project at the University of California, Santa Barbara Remote Sensing Research Unit, funded by NASA Headquarters.

A brief overview of the BROWSE testbed facility at UCSB is provided. (A more complete description can be found in Star et al, 1987). Next, some tentative algorithms for generating browseable data suitable to different disciplines are proposed. In addition, several scenarios for browsing, suggested by our scientific collaborators, are outlined to suggest the complexity of reaching a consensus in a definition. Then some future research directions are discussed. The main focus of next year's effort will be in the evaluation phase. Our hope is that public discussion of these issues will not only improve the effectiveness of the BROWSE testbed, but will also stimulate creation of browsing utilities on other spatial information systems as well.

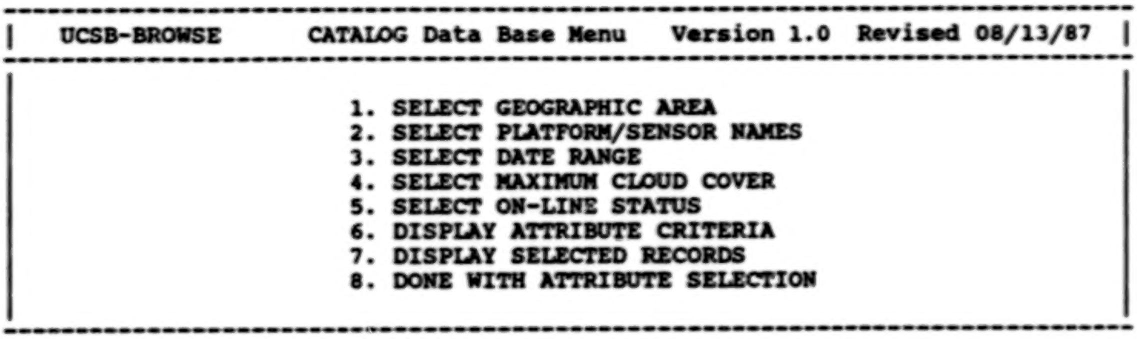
2. DESCRIPTION OF THE BROWSE TESTBED FACILITY

The first year of research, recently ended, saw the development of a rapid prototype of a browse utility at UCSB. Our objective was to get an operating version on-line and accessible via phone line to a set of collaborators from many disciplines and institutions around the United States. In the current year, we plan to incorporate their feedback on the usefulness of the system for their respective needs. What follows is a very brief description of the testbed facility.

The host facility consists of a MicroVAX II Workstation with peripheral tape drive, two 319 megabyte hard disks, a VT260 terminal, and a PC-AT as a workstation. Resident on the host computer are the database management system (DBMS), the user interface program, two databases, and a set of preprocessed browseable images. The DBMS used is a public-domain package, Relational Information Management (RIM), also being used by the NASA Ocean Data System (NODS) operated by the Jet Propulsion Laboratory.

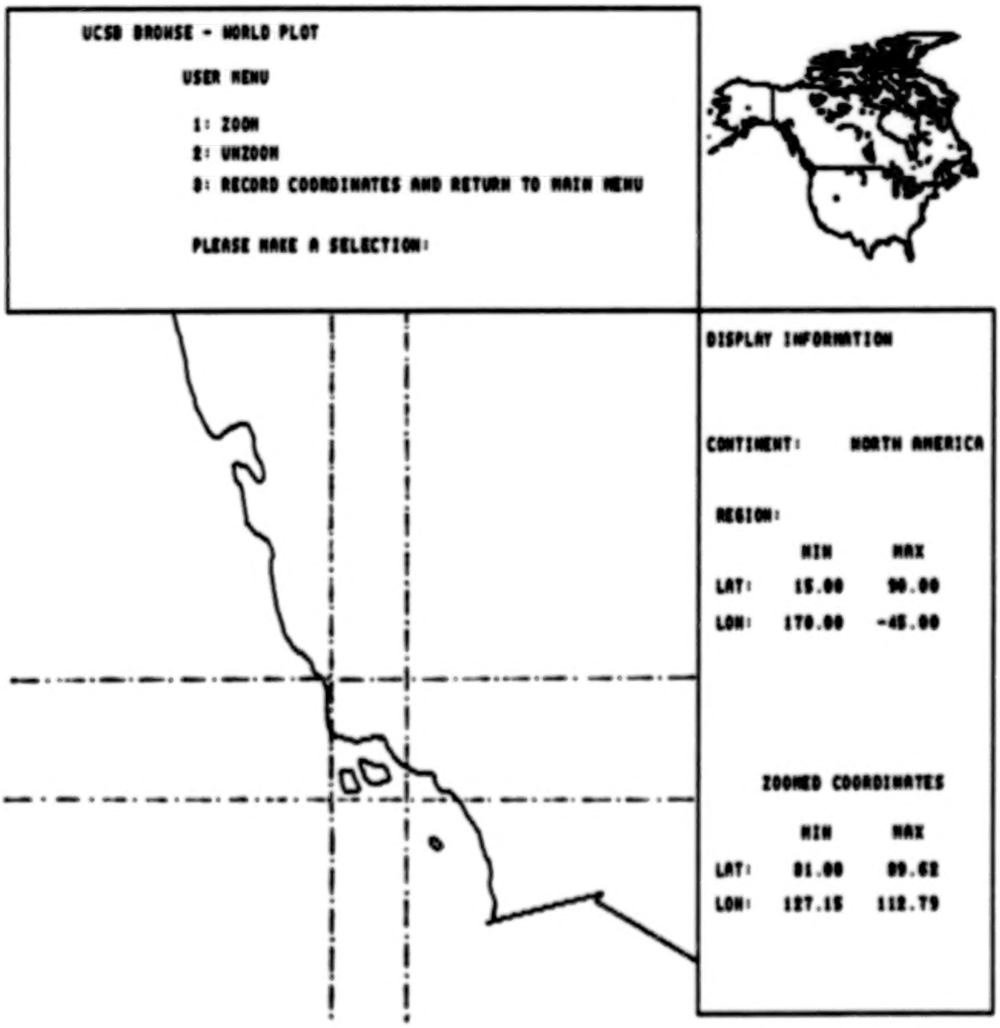
To save the user from having to learn RIM syntax or the details of the databases, we have written a menu interface that prompts the user for query attributes. The Catalog data base contains detailed records of individual images. As seen in the Catalog menu in Figure 1, the user can restrict the search to specific geographic area, sensors, dates, maximum cloud cover, and browseable status. The graphic mode for identifying a geographic search space is shown in Figure 2, using Santa Barbara, California as an example. The program then automatically formulates the appropriate database query. The current query attributes or the current set of records retrieved can then be displayed. Figure 3 shows an example of records retrieved for the same area as shown in Figure 2. Realize that at this point in the session, only information about the data is provided, not imagery. This aspect is similar to a computerized literature search at a library or the type of query the EROS data center has supported for several years (USGS, 1980).

When a Catalog database record sounds useful, the user invokes KERMIT, a public domain communications package, to transmit the preprocessed image to their local terminal. This low resolution version can then be displayed locally using an image display program written for the IBM PC-AT workstation. Using multiple windows, the display includes a single band image, its histogram, and statistics. Currently, the display uses a level slicing routine to divide the grey values into eight colors, which can be adjusted through contrast enhancement functions in the program. Figure 4 shows a black and white reproduction of the BROWSE image corresponding to the database record of Figure 3. At this point,



```
-------------------------------------------------------------------------
|  UCSB-BROWSE      CATALOG Data Base Menu   Version 1.0  Revised 08/13/87 |
-------------------------------------------------------------------------
|                                                                       |
|                    1. SELECT GEOGRAPHIC AREA                          |
|                    2. SELECT PLATFORM/SENSOR NAMES                    |
|                    3. SELECT DATE RANGE                               |
|                    4. SELECT MAXIMUM CLOUD COVER                      |
|                    5. SELECT ON-LINE STATUS                           |
|                    6. DISPLAY ATTRIBUTE CRITERIA                      |
|                    7. DISPLAY SELECTED RECORDS                        |
|                    8. DONE WITH ATTRIBUTE SELECTION                   |
|                                                                       |
-------------------------------------------------------------------------
```

Figure 1. Catalog Menu for BROWSE
Showing Query Attributes for Restricting Search



```
UCSB BROWSE - WORLD PLOT

        USER MENU

         1: ZOOM
         2: UNZOOM
         3: RECORD COORDINATES AND RETURN TO MAIN MENU

         PLEASE MAKE A SELECTION:
```

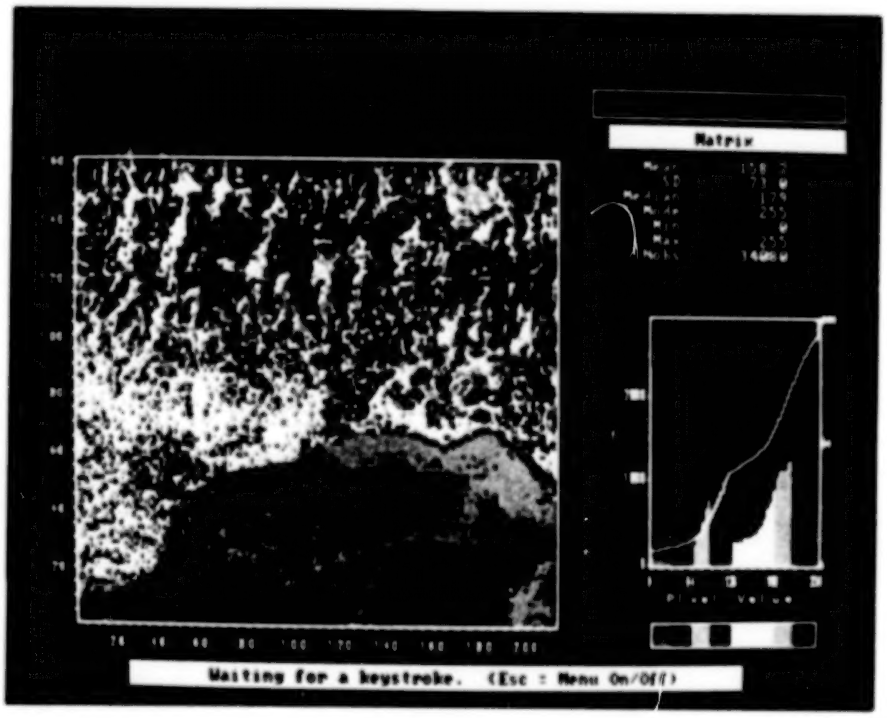
```
DISPLAY INFORMATION

CONTINENT:     NORTH AMERICA

 REGION:
          MIN      MAX
 LAT:    15.00    90.00
 LON:   170.00   -45.00

      ZOOMED COORDINATES
          MIN      MAX
 LAT:    31.00    39.62
 LON:   127.15   112.79
```

Figure 2. Example of Graphic Mode for Specifying
Geographic Region Around Santa Barbara, California

BEST COPY
AVAILABLE

```
  UCSB-BROWSE: Current number of records selected =     3

#    SCENE ID          CLD%     DATE     PLATFORM MISS SENSOR   ON-LINE?
---- ---------------   ----  --------    -------- ---- ------   --------

  1) Y5092817550X0PC2   20   09/15/86   LANDSAT  5     TM         ON
  2) Y5092817550X0BN3   20   09/15/86   LANDSAT  5     TM         ON
  3) Y5092817550X04/3   20   09/15/86   LANDSAT  5     TM         ON

End of current list of records

Would you like more details on any image? [Y/N]: y

Enter the number of the record you wish to see, between 1 and     3: 1

 1)  SCENE ID: Y5092817550X0PC2     DATE: 09/15/86     ON-LINE? ON
     PLATFORM: LANDSAT    MISSION: 5     SENSOR: TM
     CLOUD COVER:  20%     QUALITY: 5555555     HISTORY: PC2,SUBSET
     FORMAT: DIGITAL       PHYSICAL DESC: NULL
     FILE POINTER: [.TM]SANTAB86.PC2
```

Figure 3. Example of Data Retrieved from the Catalog for Santa Barbara, California



Waiting for a keystroke. (Esc : Menu On/Off)

Figure 4. BROWSE Display of Scene of Santa Barbara, California Corresponding to Record in Figure 3
(original display is in color)

the user can make a more informed decision on the suitability of the data for a spatial data processing application than from the data record alone. What makes BROWSE unique is the capability of both locating and displaying spatial data at a remote terminal.

An alternative menu path guides the user through the Directory database containing addresses and collection emphasis of various data centers around the world. Therefore, if suitable imagery is not available in the relatively small collection at the UCSB Remote Sensing Research Unit, the user can be directed to other likely sources. We plan to add Catalogs for some of these other archives to the BROWSE network. A third database of user information is planned for the coming year. This User database will permit a researcher to find other investigators who are knowledgeable about particular geographic regions, sensors, or disciplines.

3. PROPOSED BROWSING FORMATS

From the description of the testbed facility, browsing in this context can be very generally defined as locating spatial data and viewing it in some graphical form. This is not unlike how a researcher browses in a library--going to the shelf of relevant books, selecting some likely candidates, looking at the abstracts (i.e., a low resolution version of the entire text), rejecting some, finding new suggestions in the list of references, and often making serendipitous discoveries in material they were not originally aware of. It is this sense of discovery and exploration that can add Naisbitt's sense of 'high touch' to the high technology of computerized data retrieval.

Each of the NASA pilot data systems are implementing some form of browse capability. NODS includes a browse facility with data tabulations, graphics, and image formats. The user chooses a format and is presented with a list of names of browse files to view (Jet Propulsion Laboratory, 1986a). These files contain preprocessed examples of each format. Browse images of 4500 Coastal Zone Color Scanner and Advanced Very High Resolution Radiometer scenes for the West Coast Time Series have been recently put on line. At the Pilot Climate Data System, browsing includes looking at image data, but it also refers to looking at descriptions of sensors and climate parameters (NASA Goddard, 1986). Browsing or 'quick-look' support is also specified in the functional requirements for the Pilot Land Data System. Significantly, browsing is included under the data management requirements rather than those for analysis software (NASA, 1986). Recognition of this role underscores the point that browsing is a function to aid in data selection and management and should not supplant or duplicate GIS or image processing analytical functions. The Planetary Data System, is using videodisks for electronic browsing of planetary images (Jet Propulsion Laboratory, 1986b). A browse utility for EOS data is also specified (Broome, 1986).

Other data systems outside NASA are also providing browsing functions (Kubo, 1986). However, little detail of what these 'quick-look low resolution' views should show is described in any of these systems.

Among our objectives in the BROWSE project is to determine if there is a basic set of processing algorithms to generate suitable browse images. At the end of the project, we intend to provide NASA with a matrix of algorithms preferred by each discipline and suggest a small subset of these algorithms that present data in ways the majority of researchers require. The challenge is to satisfy scientists in fields as diverse as oceanography, forestry, and meteorology. The difficulty can be illustrated by considering clouds as viewed by a spaceborne multispectral scanner. To many specialists, clouds are a hindrance to information extraction. Therefore, extensive cloud cover is a negative attribute to be avoided or masked from the data. For some meteorological applications, however, it is precisely the clouds that are significant.

Providing low resolution browse data presumably involves compression in the spatial, spectral, radiometric, or temporal domains, i.e., subsampling the original data. Data compression offers several benefits in an information system and one major drawback. On the positive side, compression reduces the data storage and transmission loads on the system. Although storage capabilities are increasing (and costs decreasing) in the computer industry, the impact of EOS's proposed 262 channel High Resolution Imaging Spectrometer (HIRIS) will still be enormous in the foreseeable future (Broome, 1986). The projected data rate from HIRIS translates into one 6250 bpi density tape every second. The disadvantage of compression is that, after a certain level of reduction, the process is irreversible. Just as an abstract cannot be transformed into the original book, a highly compressed image cannot be converted by the user into the raw data. In other words, BROWSE only supplies the abstracts, as it were, so that the user can decide whether to buy the whole book.

The simplest approach to image compression is to discard all but a single spectral band. For purposes of browsing, a single band is often adequate for determining whether image data is suitable for input and analysis in a GIS. A visible wavelength band is appropriate for revealing the effects of clouds and atmospheric disturbance, for determining the presence of man-made objects, or for penetrating water surfaces. Other wavelengths have their own best uses.

A second approach capitalizes on the known relationships between spectral bands by using ratios of two bands. A ratio of reflective infrared over red wavelengths is commonly used in vegetative studies. The relative brightness is an indicator of biomass (Jackson, 1984). Geologists frequently use a mid infrared over near IR, a mid-IR over another mid-IR, or a red over blue band ratio to determine composition of surface materials (Crippen et al, in press).

Principal components analysis takes advantage of correlation in a dataset to transform it into a smaller number of dimensions. Significant scene variance may be conserved in a few synthetic output channels (i.e., components), while uncorrelated noise is collected in lower order components that can be discarded in many circumstances. We hypothesize that a single component will show enough information for a browser. The disadvantage of principal components analysis is that it can be very time consuming to compute. The transformation coefficients must be recalculated for every image, and the components may be difficult to interpret since they are scene dependent.

Linear combinations are another way to transform image data, using a predefined set of coefficients. Each combination makes a specific rotation of data in spectral space to optimize a particular feature. Perhaps the best known of this type is the Tasseled Cap transformations for Landsat Multispectral Scanner data emphasizing soil brightness and vegetation greenness (Kauth et al, 1976). A similar transformation has been developed for Thematic Mapper data (Crist et al, 1984). In essence, the first transformed axis is an ordination from wet to dry soil, while the second axis indicates biomass as a distance from the bare soil baseline. Coefficients can be easily derived for other sensors but the interpretation of the axes may be unique for each type (Jackson, 1983).

4. SOME BROWSING SCENARIOS

It is impossible to predict all the ways researchers will want to use the BROWSE facility. The collaborators who have assisted us by describing their browsing needs tend to be experienced in remote sensing applications. The new constituency NASA hopes to reach is harder to identify, and new users may have difficulty articulating what they need, other than that the system be helpful and flexible. This section describes a few of the scenarios which our multidisciplinary collaborators have identified as the process they might use in gathering GIS data.

As we progress into the age of global science, researchers are collecting datasets for large regions at high resolution. An analyst working at this scale might need to see a low resolution, single band view of many adjacent scenes acquired in a particular season by the same sensor under relatively clear atmospheric conditions.

Research projects sometimes involve multistage sampling, with low resolution data of large areas and higher resolution for selected samples. A browse facility could determine if data of all required types were available and suitable at the study site. Alternatively, when multisensor data is required, the DBMS could identify sites with all the required types. By browsing through these sites, the analyst could select the most suitable for further investigation and commitment of staff and funds.

Discipline specialists new to remote sensing may not know the capabilities of different sensors. In an extreme example, an investigator wanting to census elk populations over a large region may believe that a single scene of a low resolution sensor like the Advanced Very High Resolution Radiometer (AVHRR) would be adequate. A quick-look at a browse image should convince them immediately that AVHRR data in not appropriate for the task.

Misplacing research data over time as students come and go is an embarrassing fact of life in many academic departments. It is hopeless, as we ourselves have discovered, to go through a storeroom full of poorly labeled tapes, trying to find a valuable dataset someone in the past has painstakingly assembled. If a browse image of each dataset were available, along with a trail to the dataset itself, researchers might spend less time recreating data.

5. FUTURE DEVELOPMENT PLANS

The first year of development focused on rapid prototyping of the BROWSE testbed facility and exploring what 'browse' means to different disciplines. The existing prototype consists of a single node at UCSB with a relatively small database. In the coming grant year, we intend to expand the system to include at least one or two additional nodes with different types of image data. To accomplish this involves developing a high level activity manager that queries the appropriate databases around the network, transparently to the user, so a person does not need to learn multiple query languages.

The prototype facility has now become accessible via standard phone lines to our collaborators. We will be monitoring their use and incorporating their feedback on the user interface and the types of browse formats. In concert with this, we will be examining data compression in greater detail. Tradeoffs between storage overhead of preprocessed image data versus time delays of processing on demand will be assessed.

The existing prototype assumes the user can select appropriate sensors and will recognize when data are suitable. Since NASA hopes to establish a wider constituency for data acquired in space, more expert guidance will be necessary to aid new users in formulating appropriate queries. Development of an expert system is planned that can give intelligent advice on query attributes. The user need respond only with the parameters of their task i.e., the view that is more familiar to them. The initial expert enhancement will be in the domain of vegetation applications where a substantial knowledge base can be assembled from the research literature.

6. CONCLUSIONS

A capability to browse spatial data will be an important function in future spatial data systems. Researchers will browse in order to locate suitable input data, to recommend imagery for permanent archiving, and to request additional data acquisitions from ongoing missions. When EOS begins generating massive volumes of data in support of global science, the need for these three functions will be even greater. The BROWSE project outlined here indicates one effort by NASA headquarters to prepare for the EOS era.

As defined above, 'browsing' involves both locating and viewing spatial data. The browse function is most appropriate as a data management tool. We do not see it as a replication of existing GIS or image processing analytical functions. Several methods of compressing data into browseable format have been discussed. Additional methods, from disciplines different from those of the authors, may be identified during the testing and evaluation phase. Planned enhancements should make BROWSE more effective in serving a wider constituency of users, which is one of NASA's objectives. To be effective, a browsing function should also be fast, flexible, and friendly.

Persons interested in joining the list of BROWSE users/testers are encouraged to contact the authors at the above address. Together we can give browsing for spatial data a degree of 'high touch' comparable to the level of high tech provided by computerized database management.

Acknowledgments

The authors would like to acknowledge the contributions of Dr. Tony Villasenor and Dr. Jim Dodge at NASA Headquarters, Dr. Chuck Klose, Pat Hogan, and Iveory Johnson at the Jet Propulsion Laboratory, and those numerous individuals who are providing us with much useful input and data. Research for this project was sponsored under NASA grant NAGW-987.

References

Broome, Jr., Douglas R., "The Earth Observing System: A Multidisciplinary System for the Long-Term Acquisition of Earth Science Data from Space," *Monitoring Earth's Ocean, Land, and Atmosphere from Space--Sensors, Systems, and Applications*, vol. 97, pp. 797-822, 1986.

CODMAC, *Data Management and Computation*, 1: Issues and Recommendations, 167 pp., National Academy Press, Washington, D.C., 1982.

CODMAC, *Issues and Recommendations Associated with Distributed Computation and Data Management Systems for the Space Sciences*, 111 pp., National Academy Press, Washington, D.C., 1986.

Crippen, R.E., R.G. Blom, and J.H. Heyada, "Directed Band Ratioing for the Retention of Perceptually-Independent Topographic Expression in Chromaticity-Enhanced Imagery," *International Journal of Remote Sensing*, in press.

Crist, Eric P. and Richard C. Cicone, "A Physically-Based Transformation of Thematic Mapper Data--The TM Tasseled Cap," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-22, no. 3, pp. 256-263, 1984.

Estes, John E., "The Need for Improved Information Systems," *Canadian Journal of Remote Sensing*, vol. 11, no. 2, pp. 124-131, 1985.

Estes, J. E. and J. L. Star, "Support for Global Science: Remote Sensing's Challenge," *IEEE Proceedings*, vol. 73, no. 6, pp. 1097-1107, 1986.

Jackson, Ray D., "Spectral Indices in n-Space," *Remote Sensing of Environment*, vol. 13, pp. 409-421, 1983.

Jackson, Ray D., "Remote Sensing of Vegetation Characteristics for Farm Management," *Proceedings of SPIE*, vol. 475, pp. 81-96, 1984.

Jet Propulsion Laboratory, *Software Specification for the GOLD Catalog*, 74 pp., California Institute of Technology, Pasadena, CA, 1986a.

Jet Propulsion Laboratory, *Planetary Data System Version 1.0 System Specification*, 234 pp., California Institute of Technology, Pasadena, CA, 1986b.

Kauth, R.J. and G.S.Thomas, "The Tasseled Cap--A Graphic Description of the Spectral-Temporal Development of Agricultural Crops as Seen By Landsat," *Symposium on Machine Processing of Remotely Sensed Data*, pp. 4B-41 through 4B-51, 1976.

Kubo, Sachio, "The Basic Scheme of TRINITY: A GIS with Intelligence," *Proceedings of the Second International Symposium on Spatial Handling*, pp. 363-374, Seattle, WA, 1986.

Naisbitt, J., *Megatrends*, 333 pp., Warner Books, New York, NY, 1982.

NASA, *Pilot Land Data System Functional Requirements--Version 0.2*, 13 pp., , 1986.

NASA Goddard Space Flight Center, *Proceedings of the Second Pilot Climate Data System Workshop, NASA Conference Publication 2430*, 263 pp., National Space Science Data Center, Goddard Space Flight Center, Greenbelt, MD, 1986.

Star, Jeffrey L., Mark A. Friedl, David M. Stoms, and John E. Estes, "Browse Capability for Spatial Databases," *Proceedings of GIS'87 Conference*, San Francisco, CA, 1987.

USGS, *A User's Guide to the Teletype-Inquiry Option of the INORAC System*, EROS Data Center, Sioux Falls, SD, 1980.

BLANK PAGE

MEASUREMENT OF LONGWAVE ENERGY FROM SELECTED URBAN SURFACES USING REMOTE SENSING AND _IN SITU_ DATA

Dale A. Quattrochi

NASA, Earth Resources Laboratory
National Space Technology Laboratories
NSTL, Mississippi 39529-6000, USA

1. INTRODUCTION

A key issue in the conceptualization of cities as urban "ecosystems" is understanding the process-response mechanisms which as an integrated whole, form the urban social and biophysical environments. In construct, urban ecology applies a systematic approach to the study of cities where the links between social and biophysical interactions are examined (1). To successfully apprehend cities as urban ecosystems, therefore, requires the integrated analysis of disparate data types which reflect the synergistic interrelationships between man and his biophysical environment. Because of the complexity of these relationships, the use of geographic information systems (GIS) technology is paramount as a vehicle for aggregating and manipulating data of different types and formats. GIS methodologies are requisite for the effective modeling of the primary processes such as the hydrologic, biotic, and climatic systems, which comprise the biophysical urban environment. Moreover, to effect the cohesive development of "what if" or cause-effect scenarios which link man-land and man-atmosphere relationships, a GIS approach appears as the most direct way to segment the multiplicative factors which together produce the urban ecosystem.

1.1 The Surface-Atmosphere as a Vital Component of the City Ecosystem

One of the fundamental components that makes the city environment unique from its rural counterpart, is the climate which prevails over urban areas. The alteration of the landscape through urbanization involves the transformation of the radiative, thermal, moisture, and aerodynamic charactersitics which displace the natural channeling of energy through the solar and hydrologic systems (2)(3). Through changing the surface geometry, vegetation, and hydrology in the city-building process, man affects the local energy exchanges that take place within the atmospheric boundary layer above urban areas. It is reasonable to expect then, that one of the major results of altering the landscape through urbanization is a modification of the linkages extant within the surface-atmosphere which presides over the natural landscape.

The extreme land cover heterogeneity and spatial diversity of land cover morphology within cities precludes extensive collection of data through _in situ_ measurement techniques. It has been suggested that data on surface-atmosphere conditions could readily be acquired through remote sensing in conjunction with _in situ_ measurements (4)(5). Furthermore, the integration of remotely sensed and _in situ_ measurements with other spatial data on urban surface characteristics into a GIS format, offers an effective and efficient means for apprehending and modeling the energy exchanges which occur within the urban atmosphere. This paper, therefore, provides insight on how data from a thermal airborne scanning system can be used as an integral component within the concept of an urban ecosystem, to model interactions which occur between the land-air interface. Ultimately, these data when tied with ground level _in situ_ measurement, can be applied to GIS methods in deriving a clearer perspective of cities within an urban ecosystem context.

2.0 POTENTIAL APPLICATION OF TIMS DATA TO MEASUREMENT OF URBAN THERMAL FLUX

The Thermal Infrared Multispectral Scanner (TIMS) is a six-channel airborne sensor flown on a Lear model 23 aircraft available at the NASA, National Space Technology Laboratories (NSTL), Earth Resources Laboratory (ERL). Table 1 gives the bandwidth configuration for the TIMS sensor; a more complete description of the sensor is provided by Palluconi and Meeks (6). With a "look angle" of 39 degrees and a pixel size of 5 to 30 meters at nadir depending upon altitude above the terrain, the TIMS offers a unique potential for acquiring measurements on surface thermal energy responses. This has been substantiated by Ochoa (7) in his research on soil thermal properties, and

Table 1. TIMS Bandwidths by Channel

Channel 1	-- 8.2 - 8.6μm
Channel 2	-- 8.6 - 9.0μm
Channel 3	-- 9.0 - 9.4μm
Channel 4	-- 9.6 - 10.2μm
Channel 5	-- 10.3 - 11.1μm
Channel 6	-- 11.3 - 11.7μm

Source: (6)

by Luvall and Holbo (8) in their work with TIMS data for measurement of forest canopy temperatures. The TIMS has not until now, however, been used to measure thermal energy responses within an urban environment. It is hypothesized that the successful application of data from this scanner for deriving quantitative information on longwave urban surface energy exchange processes, will provide new information which will benefit the development of more complete models on city-atmosphere interactions.

2.1 The Salt Lake City Study: An Example

For the study addressed herein, TIMS data have been collected over a portion of Salt Lake City, Utah as a means for deriving measurements of diurnal thermal energy responses (i.e., surface heat flux) for common urban surfaces. These data have been acquired to support an analysis of surface-atmosphere interactions as a functional part of the overall urban ecosystem within the Salt Lake Valley. The primary objective of this research is to further knowledge on urban land-atmosphere longwave exchange processes by deriving measurements of responses and fluxes between day and night for urban surface thermal "building blocks". These morphological building blocks form the framework for energy exchange processes within the urban climate. They are comprised of the ubiquitous materials used in urban construction techniques, such as concrete and asphalt, along with "natural" surfaces; e.g., grass, trees, water and associated land covers. As determined by Goward (9), these basic land cover components can be categorized into three fundamental thermal classes: Pavements; Buildings; and Natural Surfaces. Hence, the focus of analysis for the present study currently under investigation, is on measuring the fluxes which occur between day and night for surfaces and construction materials innate to these three generic thermal response categories.

To support this research, TIMS data were obtained twice daily over Salt Lake City for a four day period between August 12 and August 15, 1985. These data were acquired at approximately 5:00 a.m. and 1:30 p.m. MDT for each of the four days the sensor was flown, and coincide with times of minimum surface temperature (i.e., pre-dawn) and solar noon, or the period of maximum solar irradiance (i.e., about 1:30 p.m. local time). The collection of thermal infrared data at times of minimum and maximum solar irradiance will serve as references for the comparative measurement of thermal energy flux dynamics within a summer climatic regime. TIMS data were acquired at a spatial resolution of 5 meters to facilitate the measurement of day/night thermal responses for discrete urban surfaces. From these low altitude data, six specific study sites were selected to represent a cross section of the general land cover types indigenous to the Salt Lake City metropolitan area. These study sites are typified by: 1) A portion of the Central Business District (CBD); 2) An industrial area with medium to heavy manufacturing and commercial activities; 3) An area which incorporates native dry grass vegetation near the Great Salt Lake and new residential construction into these native grasses; 4) A residential area which represents a norm for housing types and urban vegetation within Salt Lake City; 5) An urban park with large trees and heavy tree canopy closure; and 6) An urban park comprised largely of grass interspersed with a few small conifer trees. TIMS data will be analyzed for each of these six areas as a means for deriving measurements of thermal responses from common urban surface materials.

3.0 COLLECTION OF IN SITU DATA TO SUPPORT ANALYSIS OF TIMS DATA

To assist in verifying the energy responses derived from the TIMS data acquired over Salt Lake City, several forms of in situ data have also been obtained to augment analyses of the thermal remotely sensed information. In situ data collected from three different types of instruments, will be used to assess the veracity of the TIMS data and to provide other measurements related to the evaluation of urban surface-to-energy responses as a function of the overall biophysical urban ecosystem. These data also represent sources of point and continuous information which when integrated into a GIS format, will enhance both the quality and quantity of data that can be used for modeling thermal flux responses from the urban surface.

3.1 Atmospheric Information Acquired Via Sounding Balloons

One of the intervening factors which affects interpretation of thermal remotely sensed data, relates to the structure of the atmosphere between the aircraft and the surface being measured (i.e., the ground). Since all things with temperatures above absolute zero emit thermal energy, the water vapor and particulate matter content within the atmosphere at the time of data collection must be known to compensate for extraneous atmospheric thermal responses which will perturbate the accurate measurement of longwave energy from the urban surface. Consequently, important aspects of atmospheric structure such as barometric pressure, relative humidity, and height above mean terrain, must be known for specific intervals between the ground and the TIMS scanner to factor out the effects of the atmosphere on the thermal data. A more complete discussion of how the atmosphere influences interpretation of thermal remotely sensed data may be found in Anderson and Wilson (10).

To obtain data on the vertical structure of the atmosphere, meteorological sounding balloons were launched at times coincident with the day and night Salt Lake City TIMS overflights. Data from these radiosondes have been input to the LOWTRAN 6 atmospheric model (11), used at NASA/NSTL/ERL to correct TIMS data for atmospheric influences. Hence, atmospheric in situ data are important to support analysis of the TIMS data for measurement of of urban thermal energy responses.

3.2 Acquisition of Hand-held Radiometer Data

Ground-level data on surface temperatures were obtained concurrently with the TIMS overflights as a means for verifying the energy measurements, and as another source for evaluating trends in longwave energy for urban surfaces within Salt Lake City. These data were acquired using a Barnes (Model 5) Precison Radiation Thermometer (PRT-5). (Note: The use of commercial product names does not imply endorsement by NASA.) Although the PRT-5 measures thermal responses withing a single broad bandwidth (8-14μm) in comparison with the TIMS, data from this instrument are useful as a supportive information source for corroboration with thermal remotely sensed data. For example, general trends in longwave energy exchange fluxes for urban surfaces can be identified using the hand-held radiometer, and these data in turn can be correlated with observations from the TIMS as a measure of veracity. If the data from the TIMS do not show the same kinds of trends or variations as data presented by the PRT-5, this can be a clue to possible errors in the analysis or interpretation of the airborne thermal imagery. Moreover, PRT-5 data obtained for the Salt Lake City CBD currently being analyzed, show utility as an additional data source for developing a more complete apprehension of thermal energy flux for specific natural and man-made surface materials common to the city landscape.

3.3 Acquisition of Sensible Heat Measurements

As another source of ground data, a high speed thermistor attached to an automobile was used to obtain temperatures along several transects across the Salt Lake City CBD. These data will be employed as an ancillary information source to augment the results derived through analysis of the TIMS and PRT-5 data. Since it was impossible to acquire sensible heat temperatures at all points across the urban study area during the aircraft overflights, ground level temperatures were obtained with the thermistor over that portion of the city which would most likely show the greatest fluctuations in ground level temperatures; i.e., within the CBD. For analysis in the present study, these thermistor readings will provide at least a general idea of ground level temperatures extant at the time the TIMS overflights were conducted.

4.0 ILLUSTRATION OF THERMAL ENERGY RESPONSES FROM TIMS DATA

Although the derivation of quantitative energy measurements from the TIMS data for surfaces within Salt Lake City is currently in-progress, empirical examination of these thermal data provides useful insight on the process-response mechanisms which effect urban longwave energy exchanges. These data also show the potential that TIMS or similar thermal remotely sensed data have as significant input into spatial data management systems (i.e., GIS) for use by scientists in basic research, and by urban planners or other individuals who are concerned with urban environments.

4.1 Conversion of Data Into Energy Terms

Figures 1 and 2 illustrates both daytime (i.e., early afternoon) and nighttime (i.e., pre-dawn) data, respectively, from TIMS channel 3 (9.0 - 9.4 μm) for a portion of the CBD within Salt Lake City. The TIMS data used in the present study have been corrected for atmospheric perturbances using the LOWTRAN 6 model (11). Also, TIMS data in their original or "raw" format reside as digital values (dv's) and thus, must be transformed into energy terms (watts/cm2/steradian) to measure surface longwave energy responses. A function must be established, therefore, which will permit the conversion of dv's to energy values. Such a function, specific to the operation environment of the TIMS at the time of data collection is referred to as a System Transfer Equation (STE). The STE basically exists as a linear response to varying energy levels for each of the six TIMS channels, with the general form:

$$y=mx+b \qquad (1.0)$$

where

y is the recorded digital value
m is the slope of the STE (units of digital value watts/cm2/ster)
b is the intercept of the STE (units of digital values)
x is the energy being sampled (units of watts/cm2/ster)

To define m and b in equation (1.0), it is necessary to collect data on at least two points of the STE line. This task is facilitated by the presence of two blackbody sources (i.e., hot and cold) on board the TIMS, which are used as references in the data acquisition process. Actual conversion of TIMS data from dv's into energy terms is accomplished through the integration of Planck's equation, using the blackbody reference sources, to convert temperature to energy emitted by the references for a specific wavelength region. This procedure has been applied to the data depicted in figure 1 and as a result, measurements of total surface longwave energy reflected or emitted (i.e., upwelling energy) by urban surfaces can now be derived.

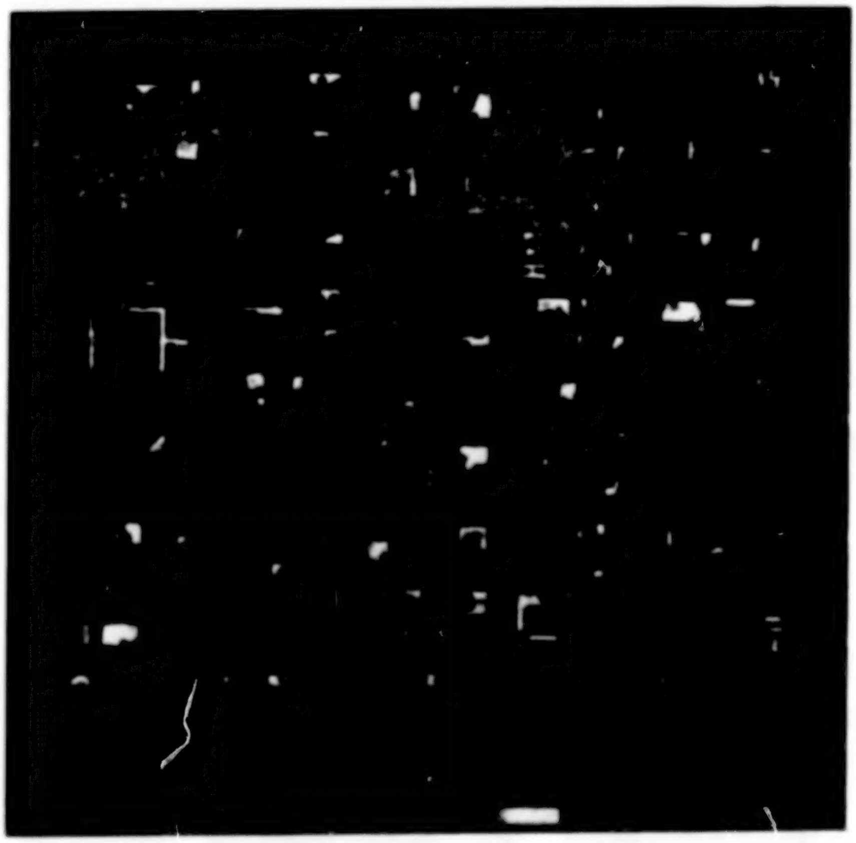


Figure 1. TIMS Channel 3 Day Data.

BEST COPY AVAILABLE

To facilitate analysis diurnal energy fluxes, the data shown in figures 1 and 2 have been registered to each other to produce an overlayed day/night data set. These data have also been georeferenced to a common map base (i.e., UTM) to assist in accurately locating discrete surface material samples for evaluating thermal energy responses. Thus, the TIMS data for Salt Lake City exist at least in concept, as an energy response data base for use in assessing the change in longwave thermal flux between day and night for various land covers.

4.2 Empirical Observations of Urban Energy Responses

As observed in figure 1, the urban surface during the daytime represents a wide array of thermal energy responses, which reflects the diverse range of thermal absorption, conduction, transmittance, and emittance capacities of man-made and natural materials within the city. Aside from streets and rooftops, building shadows are one of the more prevalent responses present within the photograph (e.g., the tall building at the bottom right of the photo). The differential heating and emittance properties of building materials and rooftops is exhibited by the Salt Palace, a large sports and convention facility located at the center left of the photograph. The concave

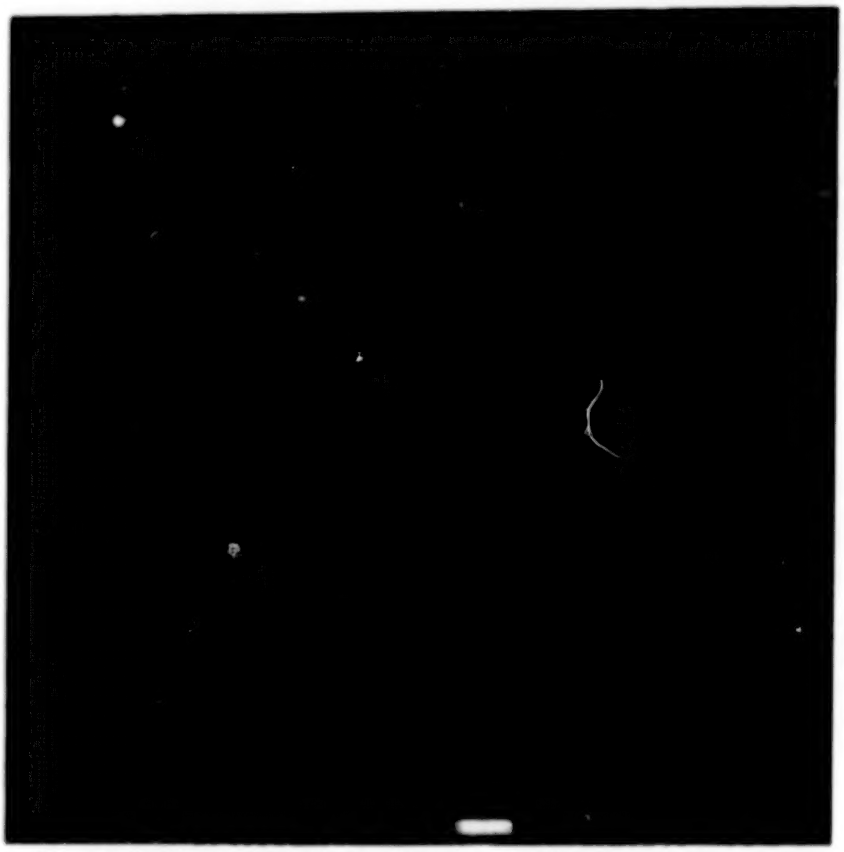


Figure 2. TIMS Channel 3 Night Data.

BEST COPY AVAILABLE

roof and various building materials on this structure illustrate the complexity of thermal energy responses from urban structures within the Salt Lake City CBD. Pavements (i.e., streets) appear relatively uniform in tone due to the ubiquitous asphalt construction of these surfaces.

Another property which must be dealt with in analysis of urban longwave energy responses is that of highly reflective surfaces such as metal roofs. Metallic surfaces because of their emmissive properties in the thermal region, reflect solar radiation and appear "cooler" than natural surfaces, such as vegetation. This characteristic is displayed by the Mormon Tabernacle, the oval-shaped structure in the top left of figure 1. Also present within the same block as the Tabernacle building, is a fairly dense vegetative canopy created by the presence of gardens and trees. Vegetative surfaces, as a response to evapotransporation and shading effects, have a significantly reduced energy response in comparison with man-made surfaces.

The TIMS night data (figure 2) depicts another perspective of energy response relationships for urban surfaces within the CBD. As noted by Goward (9), buildings appear as thin plates enclosed by a volume of air or ("hollow tubes") within the urban thermal regime; this is illustrated by the lower energy responses from building rooftops (e.g., dark responses on figure 2). Aside from buildings, however, the overall impression of surface energy responses from the night TIMS data is one of relative uniformity. It appears that the differences in energy responses exhibited during the day for specific man-made and natural surfaces is mitigated at the time of maximum cooling. This empirical observation will be investigated further in the quantiative analysis of the day and night TIMS data acquired over Salt Lake City.

4.3 Visual Enhancement of TIMS Data for Urban Thermal Energy Response Observations

Another perspective of urban surface thermal energy fluxes can be obtained from TIMS data through image enhancement. Figure 3 represents a three-dimensional view of longwave energy reflected or emitted by specific surfaces from TIMS channel 3 (9.0 - 9.4μm) day data. As seen in the figure, tall buildings appear white due to both reflection and emittance of thermal energy from the sides of the structures themselves. Areas with substantial vegetative cover or shadows may be observed as features "sunken" into the three dimensional landscape. In contrast, figure 4 represents a three-dimensional perspective of TIMS channel 3 night data for the same area.

Again, the overall appearance of surfaces at night is one of relative thermal uniformity, except for the large "sunken" feature near the center of the photo, which relates to the "hollow tube" effect created by the air column within the Salt Palace structure.

As illustrated by figures 3 and 4, image enhancement of TIMS data is useful for substantiating general trends in surface energy responses, and also for providing a different perspective of surface thermal fluxes which can be used to augment both empirical and quantiative observations. For example, different rotations of the TIMS data exhibited in figures 3 and 4 yields subtantially different perspectives of longwave energy reflected or emitted from discrete urban surfaes. In turn, these are useful for modeling how thermal energy flux is a variant of sun angle and the vertical and spatial position of various surfaces across the urban landscape. Hence, image enhancement of TIMS data can yield important empirical results which can be used to support or verify quantitative measurement of urban surface thermal energy responses.

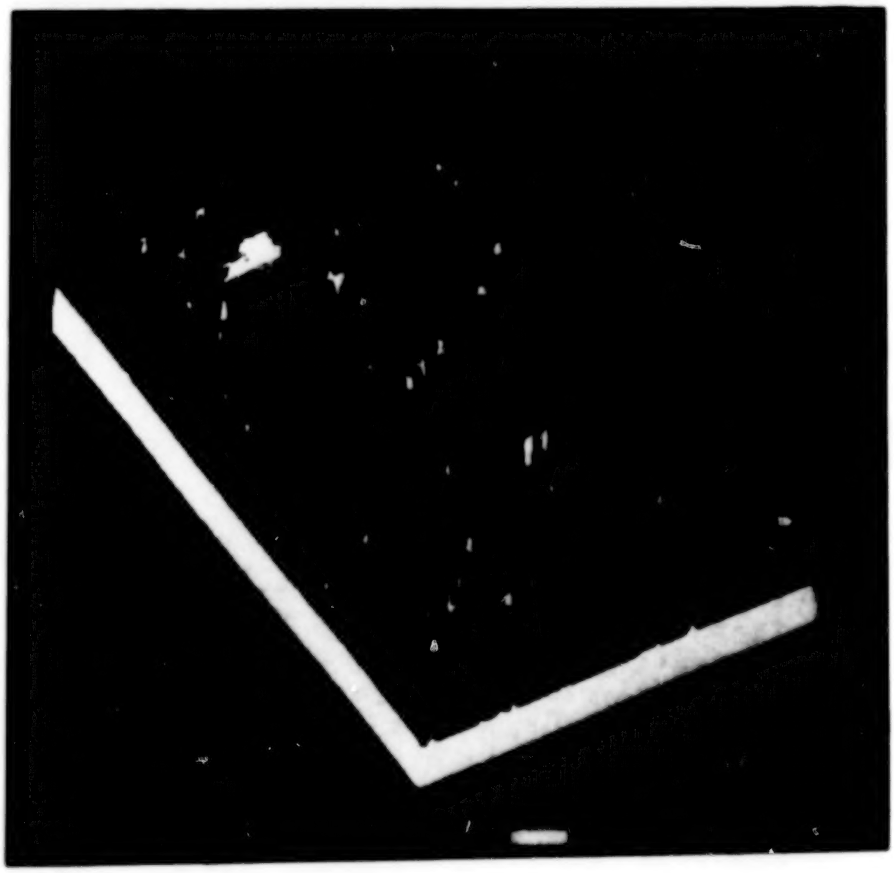


Figure 3. Three-Dimensional View - TIMS Day Data.

BEST COPY AVAILABLE

5.0 CONCLUSION

The complexity of urban longwave energy surface responses has in the past, thwarted efforts to derive definitive measurements of thermal fluxes within the city landscape. With the advent of improved remote sensors such as the TIMS, it is now appears that quantitative measurements of city-atmosphere interactions as reflected by diurnal thermal fluctuations, can become a reality. Moreover, when these remote sensing data are viewed in the context of a potential thermal energy data base as a function of an integrated GIS, their utility becomes even more enhanced.

This paper has presented a general overview of the TIMS and the application of data from the sensor for providing new and significant information on urban thermal energy process-response mechanisms. The pervading theme throughout, however, has been that remote sensing is a vehicle for the acquisition of pertinent data which can be used to construct models of urban surface-to-atmosphere relationships. The analysis of these data must be supported by adequate in situ measurements to insure that the thermal data will be correctly interpreted, and to

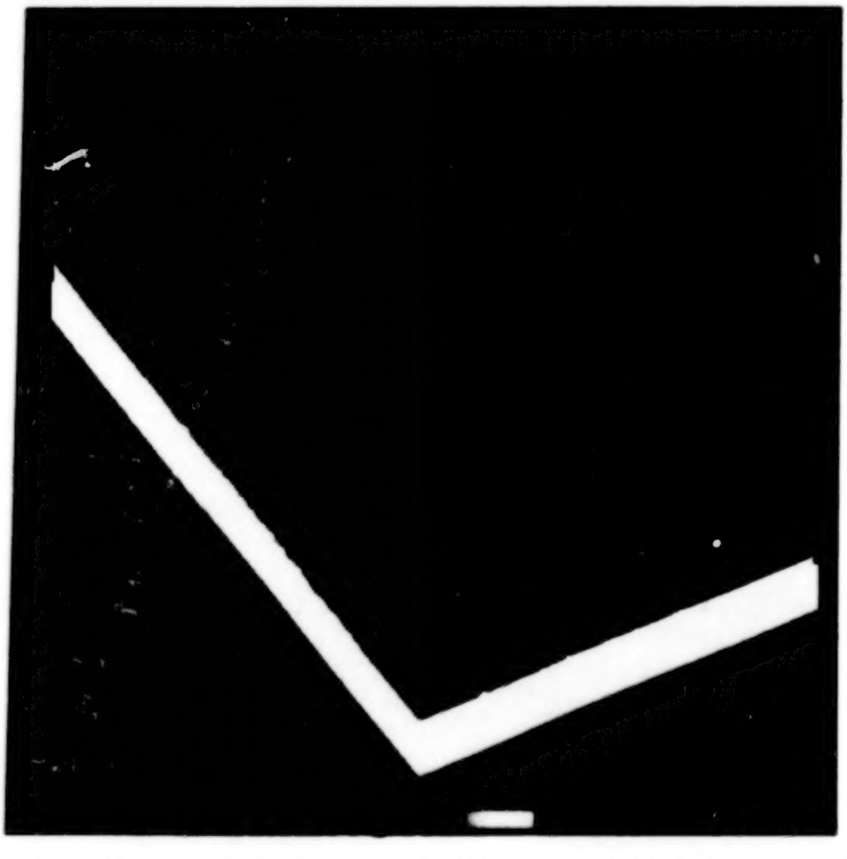


Figure 4. Three-Dimensional View - TIMS Night Data.

BEST COPY AVAILABLE

provide additional data which can be used to augment our understanding of the cause and effect operators which influence surficial thermal energy regimes.

References

1. Douglas, Ian, 1983: The Urban Environment. Baltimore: Edward Arnold. 229 p.

2. Oke, T.R., 1978: Boundary Layer Climates. New York: Methuen. 372 p.

3. Oke, T.R., 1982: The Energetic Basis Of The Urban Heat Island. Quart. Jour. Royal Meteorolog. Society, 108: 1-24.

4. Myrup, Leonard O., 1969: A Numerical Model Of The Urban Heat Island. Jour. Applied Meteorol., 8: 908-918.

5. Outcalt, Sam I., 1972: A Reconnaissance Experiment In Mapping And Modeling The Effect Of Land Use On Urban Thermal Regimes. Jour. Applied Meteorol., 11: 1369-1373.

6. Palluconi, Frank D. & Gerald R. Meeks, 1985: Thermal Infrared Multispectral Scanner (TIMS): An Investigator's Guide To TIMS Data. NASA, Jet Propulsion Laboratory, Publication 85-32. 14 p. Plus Appendicies.

7. Ochoa, Michael C., 1986: Evaluation Of Thermal Infrared Multispectral Scanner Image Data For Characterization Of Surface Soil. Unpub. M.S. Thesis, Auburn University. 90 p.

8. Luvall, J.C. & H.R. Holbo, 1987: Using The Thermal Infrared Multispectral Scanner (TIMS) To Estimate Surface Thermal Response. Proceed. International Conf. On Measurement Of Soil And Plant Water Status, Vol. 2 - Plants. July 6-10, 1987, Utah State University: 115-120.

9. Goward, S.N., 1979. The Physical Nature Of The Interface As A Factor In Urban Climate. Unpub. Ph.D. Disser., Indiana State Univer. 302 p.

10. Anderson, J.M. & S.B. Wilson, 1984. The Physical Basis Of Current Infrared Remote-Sensing Techniques And The Interpretation Of Data From Aerial Surveys. Int. Jour. Remote Sensing, 5 1: 1-18.

11. Kneizys, F.X., E.P. Shettle, W.O. Gallery, J.H. Chetwynd, Jr., L.W. Abreu, J.E.A. Selby, S.A. Clough, and R.W. Fenn, 1983: Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 6. U.S. Air Force Geophysics Laboratory, Environmental Research Papers, No. 846, AFGL-TR-83-0187. 200 p.

BEST COPY AVAILABLE

BLANK PAGE

UTILITY OF TOPOGRAPHIC DATA AND SYNTHETIC APERTURE RADAR DATA FOR FORESTRY APPLICATIONS IN MOUNTAINOUS TERRAIN

Shih-Tseng Wu

Earth Resources Laboratory
National Space Technology Laboratories
National Aeronautics and Space Administration
NSTL, Mississippi 39529-6000

ABSTRACT

Digital elevation data were used to study elevation gradients that affect tropical forest stand characteristics. The data were produced after digitizing the contour lines on a topographic map of the study area. A rubber sheeting computer program was used to georeference synthetic aperture radar (SAR) data sets to the digital elevation data. The TOPO computer program developed at the Earth Resources Laboratory (ERL) at NASA's National Space Technology Laboratories (NSTL) was used to generate the terrain slope and aspect angle data from the terrain elevation data set. A computer program also developed at ERL was then used to compute the local incidence angle of the target/scene as seen by the SAR system, using the slope and aspect data and radar look angles. The computed local incidence angles were used to delete those data areas that were shadowed (incidence angle greater than 90 degrees) and to produce the local incidence angle data that were not shadowed. The results of data analysis indicate that SAR data are related to elevation gradient. It is known that tree species, stem frequency, and tree height are related to temperature and precipitation, both of which are influenced by elevation. The results, then, imply that SAR data are related to forest structure as it changes over that elevation gradient.

1. INTRODUCTION

The electromagnetic (EM) wave interactions that take place in various earth surfaces need to be understood for future advancement in remote sensing. A synthetic aperture radar system can be built with multifrequency, multipolarization, and multilook-angle capabilities, and longer wavelength radar waves can penetrate a vegetation canopy. Due to these attributes, SAR is a useful tool in studying complex targets. Forest vegetation and some cultural targets represent important components of the earth's surface, comprising complex, three-dimensionally heterogeneous structures as compared to homogeneous grasslands and agricultural crops. Several recent studies (2, 4, 7, 8) reported the use of quantized SAR data in interpreting the radar wave interaction of forest stands. The reported results provide some understanding of how radar waves interact with forest canopies and how SAR data can be used in determining forest stand characteristics. Nevertheless, those results are limited to relatively flat areas; i.e., not over a mountainous area with substantial differences in elevation.

Since inventorying and mapping of forest resources are inevitably comprised of study areas that are mountainous and since SAR data are sensitive to local incidence angle variation, there is a need to remove the effect of slope and aspect variations to obtain the true radar signature of the targets. This study is part of a continuing effort focusing on the development of techniques on the use of a geographic information system (GIS) and spatial statistics in assessing the SAR data characteristics of forest stands in mountainous terrain, particularly to find whether SAR data are correlated with elevation variation. It is known that tree species, stem frequency, and tree height are related to temperature and precipitation, both of which are influenced by elevation. Assessing SAR data versus elevation, therefore, is the much needed first step to find whether the SAR system is capable of detecting the forest structure difference over a specified elevation gradient.

2. STUDY AREA

The study area, located in the northeastern portion of Costa Rica, is approximately 20 km southwest of the city of La Selva (Figure 1). Elevations range from 160 meters above sea level in the northeast to approximately 2330 meters in the southwest portion of the study area. The area is now part of the La Selva-Volcan Barba transect on the Atlantic slopes of the Cordillera Central in Costa Rica. This transect is a spectacular tract of undisturbed tropical forest extending from La Selva Biological Station at 30-100m to the summit of the dormant Barba Volcano at 2,900m elevation, 35km to the south of La Selva (Figure 2).

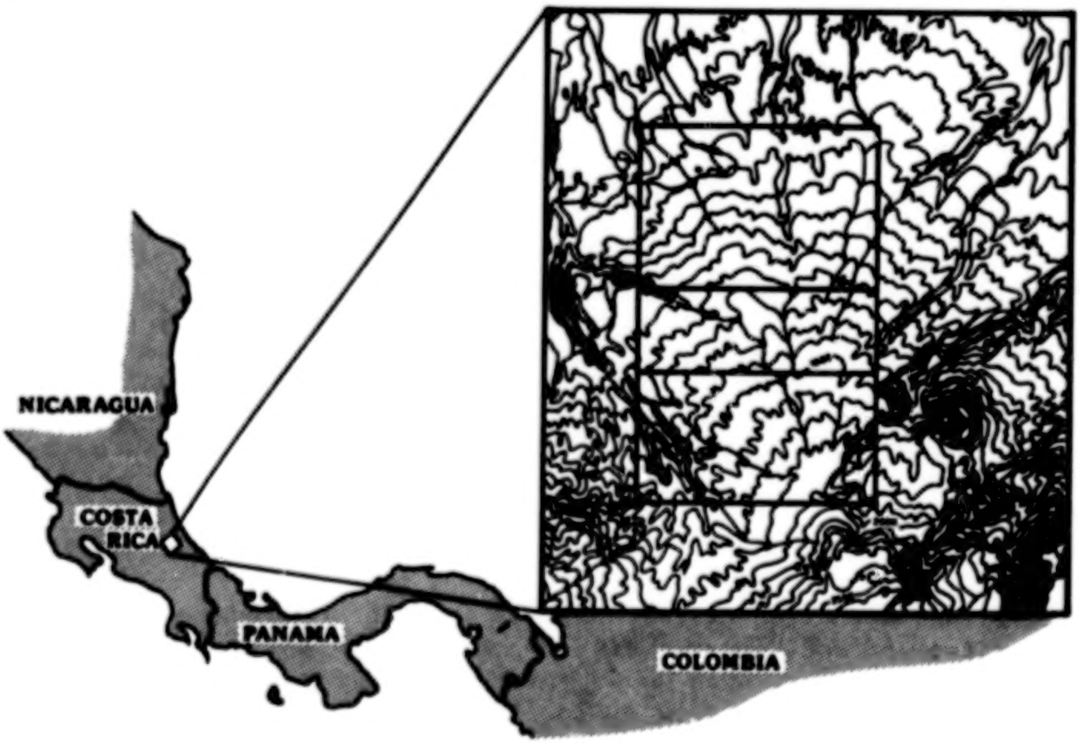


Figure 1. Tropical forest SAR study area of Costa Rica.

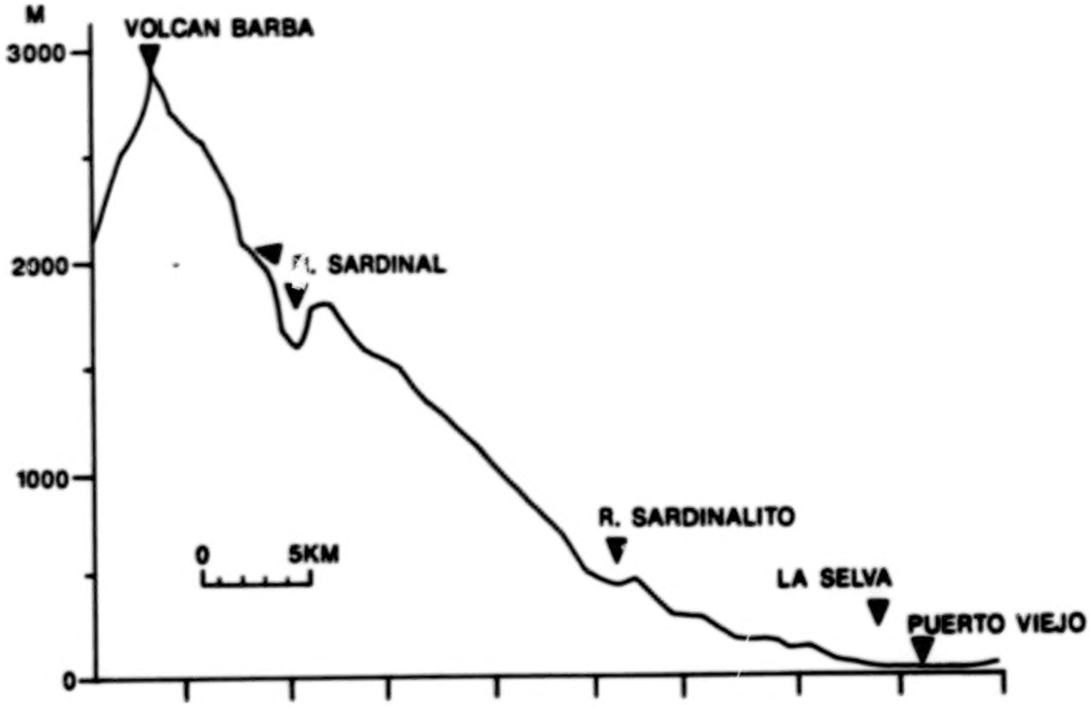


Figure 2. Profile diagram of the La Selva-Volcan Barba gradient, on the volcano; vertical scale is exaggerated.

The area of protected forest now totals over 47,200 hectares, and is believed to be the last remaining strip of undisturbed forest in Central America to cover such a wide range of elevations (Table 1).

Life Zone	Elevation (m)
Tropical Wet Forest	30 - 250
Tropical Wet Forest, Premontane Transition	250 - 600
Premontane Rain Forest, Perhumid Transition	600 - 780
Premontane Rain Forest	780 - 1450
Lower Montane Rain Forest	1450 - 2500
Montane Rain Forest	2500 - 2900

Table 1. Elevation Ranges of Life Zones as Represented on the La Selva-Volcan Barba Ecocline.

The elevation gradient passes through four life zones and two transitional zones of the Holdridge Life Zone System (3): Tropical Wet Forest; Tropical Wet Forest, premontane transition; Premontane Rain Forest, perhumid transition; Premontane Rain Forest; Lower Montane Rain Forest; and Montane Rain Forest. Detailed mapping of life zones in this area was carried out under the auspices of NSTL/ERL's tropical forest dynamics research program.

The life zones occurring on the La Selva-Volcan Barba ecocline are widely represented in Costa Rica, comprising over 45% of the national area. Costa Rica's ten forest-bearing life zones are themselves globally significant, accounting for an estimated 75% of the earth's remaining natural tropical forests. Access to this area presents an unprecedented opportunity to study diversity, species distribution patterns, and vegetation-environment interactions.

3. SAR DATA SOURCES

Airborne SAR data were acquired over the study area with the NASA Jet Propulsion Laboratory (JPL) SAR system (L-band, 1.225GHz, 24.6cm wavelength) flown in the NASA Ames Research Center CV-990 research aircraft on April 1, 1985. The SAR system (6) simultaneously recorded HH, HV, VV, and VH polarizations (H=horizontal, V=vertical). It was flown with a south-to-north flight pass with the radar antenna looking to the east (the antenna was mounted at the right side of the aircraft). SAR images were produced by post-flight optical and digital correlation. The SAR data acquisition geometry, namely aircraft altitude and across-track swath coverage, resulted in a wide range of incidence angle variation and nonuniform across-track resolution. Nevertheless, through a conversion formula, the digital data provided by JPL consisted of a uniform ground range pixel size of 10.98 meters along-track and 10.98 meters across-track. In addition, the wide range of incidence angle variation also resulted in nonuniform across-track radar return signal strength, as represented by the digital number (DN), over a uniform target.

The radar return signal strength is represented by a DN from 0 to 255; the higher the DN, the higher the radar backscatter in a relative sense. The system used sensitivity time control (STC) signals in all receiving channels to boost radar return at far range (6). The data sets for this study were corrected with removal of or compensation for all known parameters, such as the STC signals, the receiver gain settings, and the antenna gain pattern. The data are compatible with the scatterometer data without target (or absolute) calibration. Because of this correction the mean DN value at a smaller incidence angle was significantly less than that of the mean DN value at a larger incidence angle. Nevertheless, the data represent radar backscatters that can be obtained by the radar scatterometer measurement, and they should be utilized for the quantitative analysis of SAR data when available.

In the analysis of multiple incidence angle Shuttle Imaging Radar (SIR-B) data for determining forest stand characteristics, Hoffer, et al.

(2), clearly stated that incidence angle controlled, to a very large extent, the characteristics of the data and the type of information that can be obtained from L-band HH-polarized satellite SAR data. Due to this finding, no attempt should be made to correct the across-track DN variation of the aircraft SAR data, in which the incidence angle varies from 8.3 to 53.8 degrees.

The topographic SAR images, using HH, VV, and VH polarization data and the elevation data of the study area, are shown in Figures 3a and 3b.

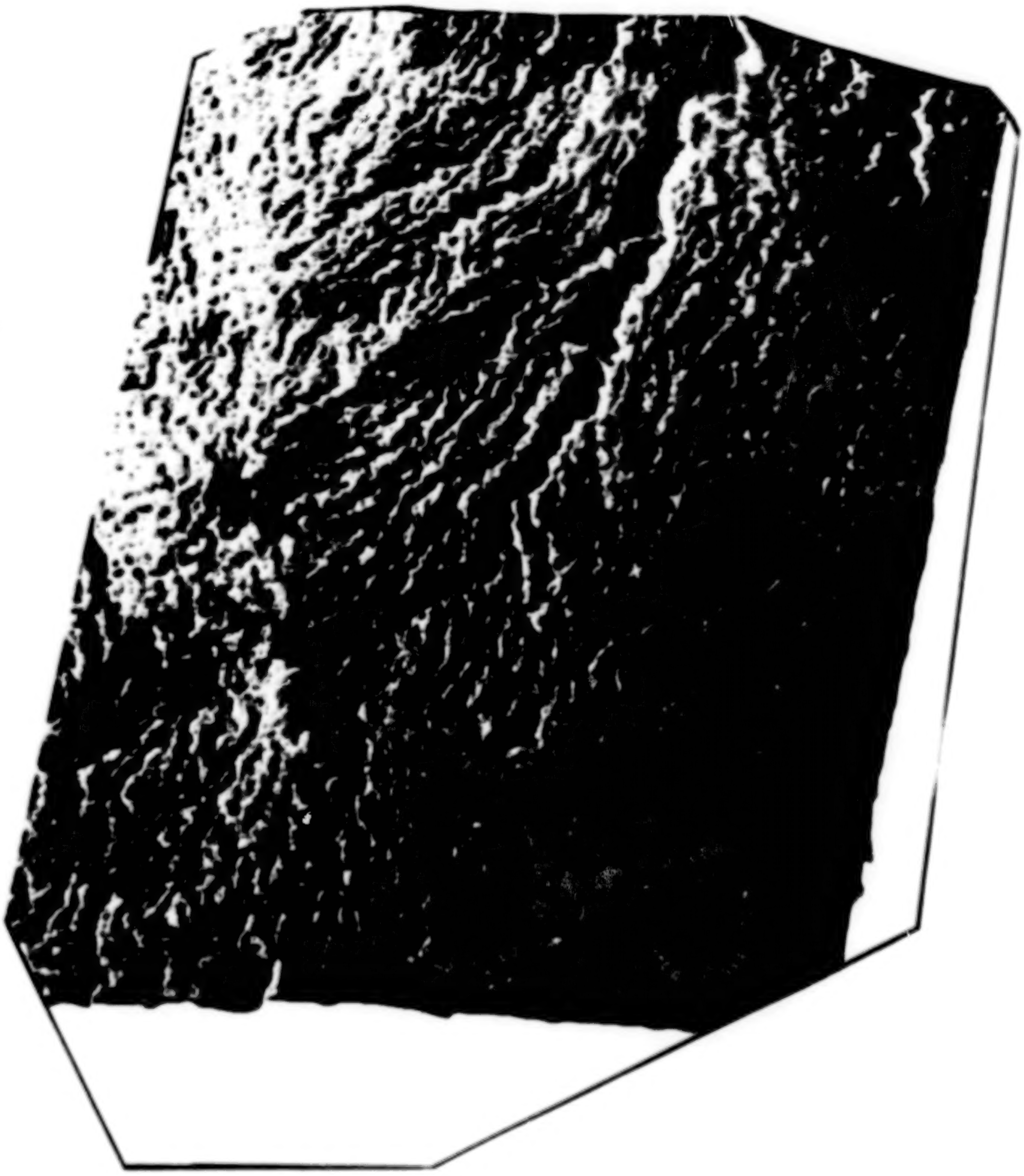


Figure 3a. Topographic SAR image of northern portion of study area.

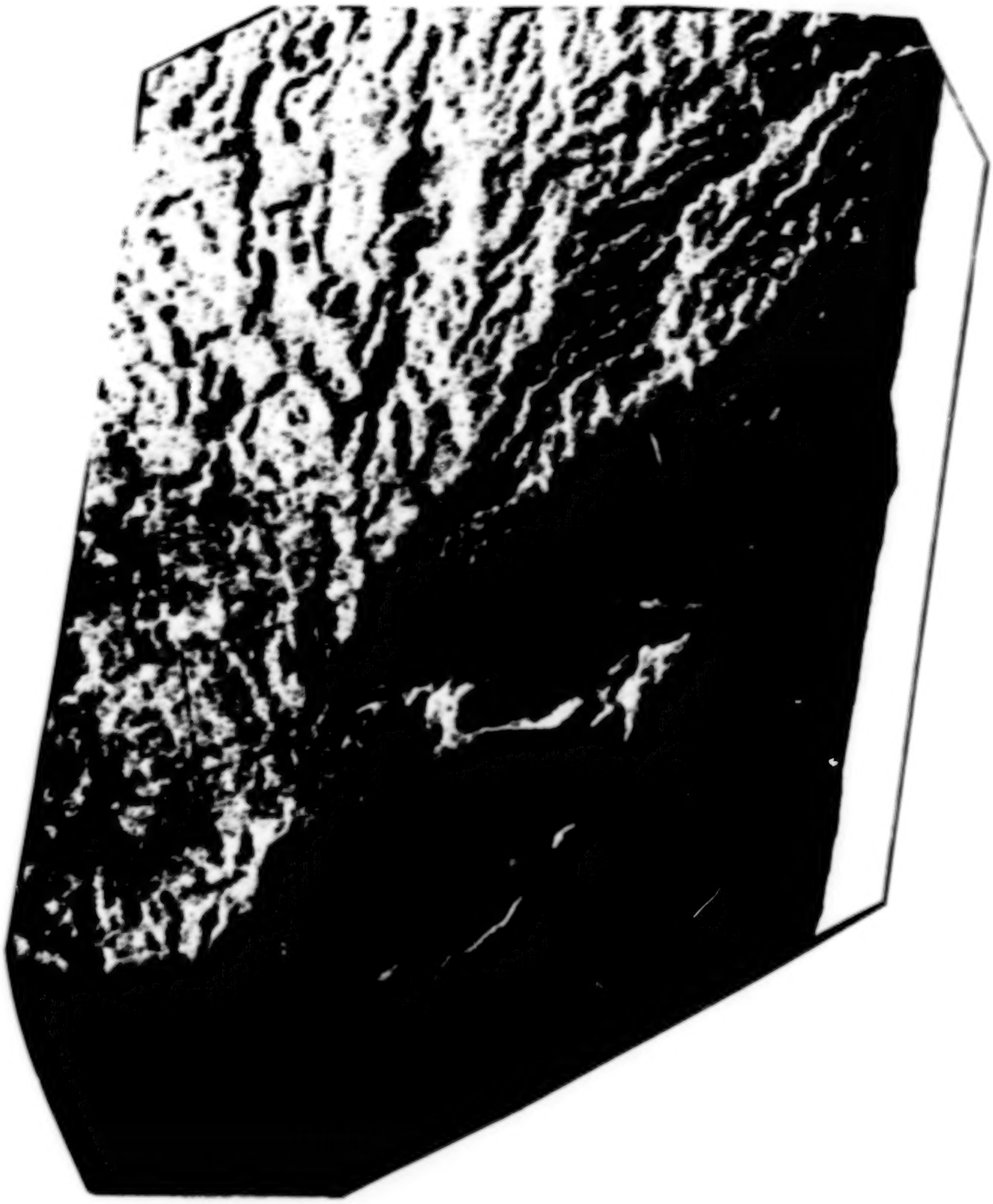


Figure 3b. Topographic SAR image of southern portion of study area.

4. TECHNIQUES

Since the data sets were acquired over a mountainous area with elevations varying from 160 to 2330 meters, since terrain slope and aspect angle affect true local incidence angle as seen by the SAR system, and since SAR data are sensitive to incidence angle variation, it is necessary to integrate elevation data with SAR data to compute the true per-pixel local incidence angle as seen by the SAR system.

BEST COPY
AVAILABLE

The digital elevation data were produced after digitizing the contour lines on a topographic map of the study area. A rubber sheeting computer program was used to georeference the SAR data sets to the digital elevation data. After georeferencing, an integrated data set containing SAR and elevation data was produced with satisfactory registration accuracy, verified by low rms error of registration and visual analysis of the three-dimensional display of SAR images. The elevation data of the integrated data set were then used to obtain the slope and aspect angle data, through the use of TOPO, a computer program developed at NSTL/ERL. A computer program also developed at ERL was then used to compute the per-pixel local incidence angle of the target/scene as seen by the SAR system, using the slope and aspect angle data and the radar look angle. If the local incidence angle is greater than 90 degrees, the pixel is not illuminated by the SAR system (radar shadowing produced by steep slope). It is necessary to exclude those data points that are shadowed in SAR data analysis. For those data points that are not shadowed, the local incidence angles are computed and applied. Figure 4 illustrates the techniques employed in this study.

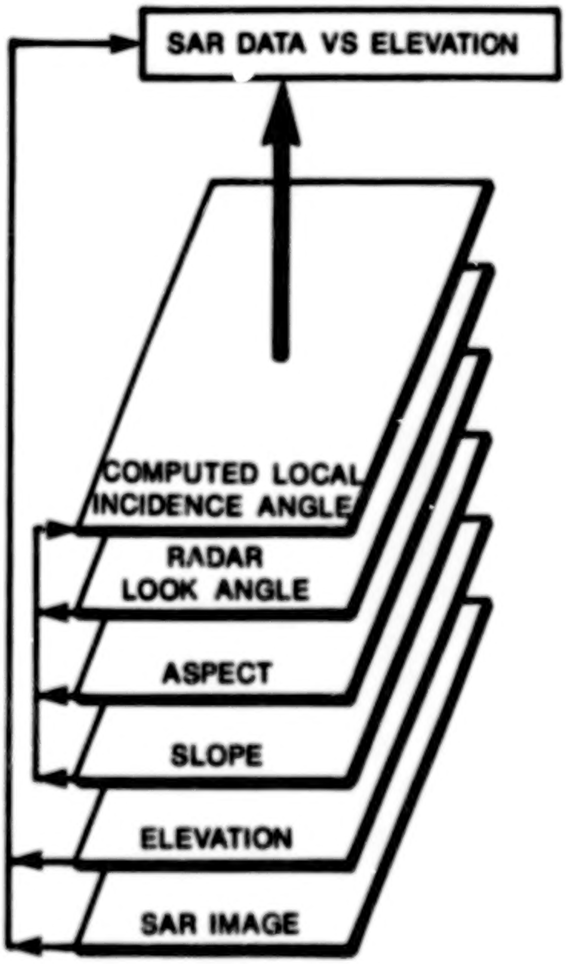


Figure 4. The use of GIS and spatial statistics in assessing SAR data versus terrain elevation.

5. RESULTS AND DISCUSSION

Due to SAR data sensitivity to incidence angle variation, it is necessary to limit the data area to a narrow strip with incidence angle values ranging from 35 to 55 degrees (7). The data strips are those of Figures 3a and 3b. A random number generator is used to select 750 points from the elevation gradient, ranging from 160 to 2330 meters. A per-pixel local incidence angle computing program is then applied to delete those data points with incidence angles less than 35 degrees or greater than 55 degrees, resulting in 314 sample data points for regression analysis.

Linear regression analysis techniques (5) are used to compute the sample correlation coefficients (r), using the elevation data as the independent variable and SAR data as the dependent variable. The computed r-values of four-polarization data, using 314 randomly selected and incidence-angle-corrected test points, are given in Table 2.

	SAR Polarization			
Est. Parameters	HH	HV	VV	VH
Sam. Corr. Co. r	-0.19	-0.28	-0.26	-0.25
Intercept a	117	125	119	124
Slope ($\times10^{-2}$) b	-0.89	-1.39	-1.26	-1.24
2-Tail Signif.	0.001	0	0	0

Table 2. Parameters of Estimation Using SAR Data Vs. Elevation

Also included in the table are the a- and b-values of the predicting equation:

$$Y = a + b * X,$$

where X is the independent variable;
Y is the dependent variable.

To illustrate the results of data analysis, the 314 sample points and the predicting equation of HV polarization are shown in Figure 5.

In summary, the results of data analysis indicate that SAR data, HV polarization in particular, are related to elevation gradient. It is known that tree species, stem frequency, and tree height are related to temperature and precipitation, both of which are influenced by elevation. The results, then, imply that SAR data are related to forest structure as it changes over that elevation gradient. Future research would then be focused on developing techniques to detect and assess differences in the structure of forests occurring in mountainous areas with substantial differences in elevation.

BEST COPY AVAILABLE

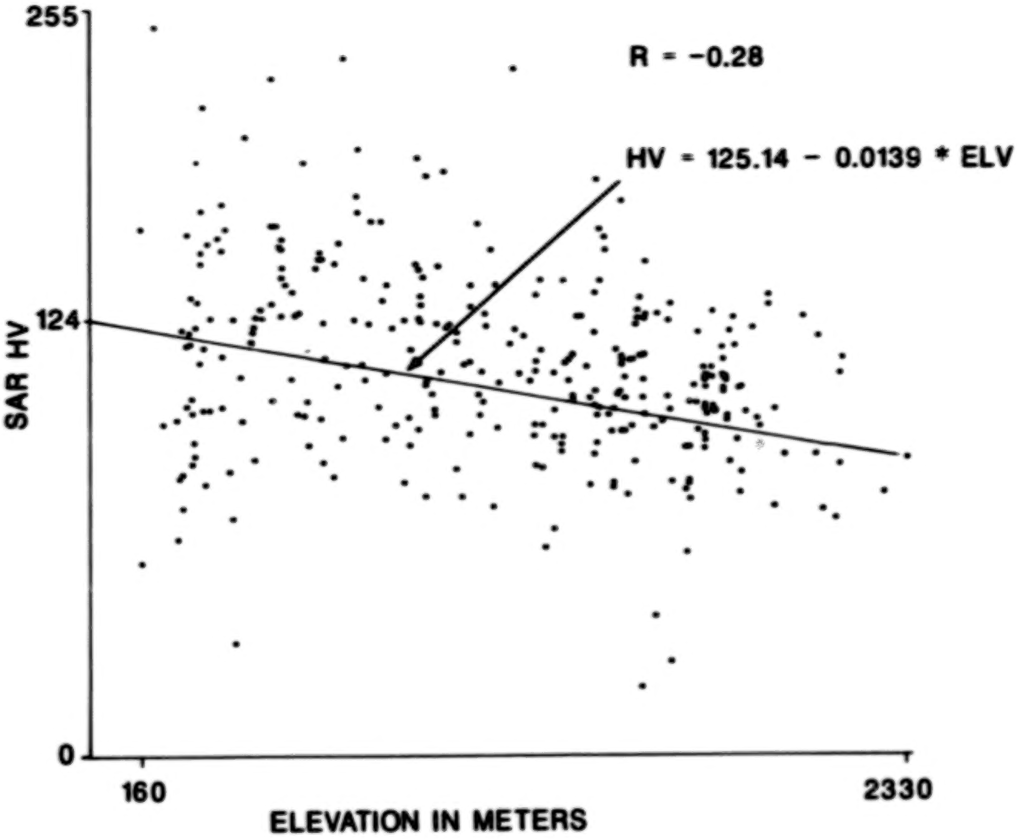


Figure 5. HV polarization SAR data vs. elevation (ELV) of La Selva tropical forest study area.

References

1. Graham, M.H., B.G. Junkin, M.T. Kalcic, R.W. Pearson, and B.R. Seyfarth, 1987: Earth Resources Laboratory Applications Software (ELAS): NASA NSTL Earth Resources Laboratory, Report no. 183.

2. Hoffer, R.M., D.F. Lozano-Garcia, D.D. Gillespie, P.W. Mueller, and M.J. Ruzek, 1986: Analysis of Multiple Incidence Angle SIR-B Data for Determining Forest Stand Characteristics: Paper presented at the Second Spaceborne Imaging Radar Symposium, Jet Propulsion Laboratory, Pasadena, California.

3. Holdridge, L. R., 1967: Life Zone Ecology, Tropical Science Center, San Jose, Costa Rica, 206pp.

4. Sieber, A.J., 1985: Forest Signatures in Imaging and Non-Imaging Microwave Scatterometer Data: ESA J., vol. 9, pp. 431-448.

5. Sokal, R.R. and F.J. Rohlf, 1981: Biometry, W.H. Freeman, San Francisco.

6. Thompson, T.W., 1983: A User's Manual for the NASA/JPL Synthetic Aperture Radar and the NASA/JPL L- and C-Band Scatterometers: JPL Publication 83-38, p. 38.

7. Wu, S.T., 1987: Potential Application of Multipolarization SAR for Pine Plantation Biomass Estimation: IEEE Transactions on Geoscience and Remote Sensing, vol. GE-25, no. 3, pp. 403-409.

8. Wu, S.T. and Steven A. Sader, 1987: Multipolarization SAR Data for Surface Feature Delineation and Forest Vegetation Characterization: IEEE Transactions on Geoscience and Remote Sensing, vol. GE-25, no. 1, pp. 67-76.

Remote Satellite Imaging and U.S. Information Policy:

National Security vs. International Competition

Mark E. Brender
ABC News
Washington, D.C.

and

Manley R. Irwin
University of New Hampshire
Durham, N.H.

Association of American Geographers
Washington D.C.
November 18, 1987

I. Introduction

U.S. remote sensing satellites are currently beset by two opposing forces. On the domestic side, private operators are subject to licensing requirements by the National Oceanic and Atmospheric Administration (Department of Commerce), the Department of Defense, and the Department of State. Regulation has centered on the question of national security and foreign policy concerns.

On the international side, U.S. remote sensing satellites are experiencing intensified competition in both satellite scanning, imaging enhancement and launching facilities. As one looks to the future, the tension between domestic regulation and foreign competition spills over into several policy questions.

> First, will national security regulation handicap the ability of U.S. firms to compete for remote imaging services within the United States?
>
> Second, will national security regulation inhibit the ability of U.S. firms to compete for customers in overseas markets?
>
> Third, can U.S. firms, satellite or launching, effectively compete in price and cost against their overseas counterpart?
>
> Fourth, will the U.S. find it necessary to invoke export control laws that govern the selling of satellites or components to overseas customers who in turn deal with the communist bloc?
>
> Finally, will the U.S. find it necessary to extend national security regulations to NATO and non-NATO entities in the field of remote sensing?

Stated differently, will the U.S. adopt the COCOM model of Europe and Japan in the area of remote sensing exports?

Before exploring these policy issues, we will examine the evolution of regulation of private satellite sensing systems in the U.S. and then identify the evolution of foreign competition in remote satellite sensing.

II. Evolution of Regulation

History

From 1972 to 1984 the U.S. government enjoyed a virtual monopoly in commercial remote satellite sensing. Remote sensing began as the Earth Resources Technology Satellite, or ERTS, later renamed Landsat. In 1979 remote sensing systems were transferred from NASA to the National Oceanic and Atmospheric Administration, Department of Commerce as a prelude to privatization under the Carter administration. In 1984 Congress passed the Land Remote Sensing Commercialization Act which privatized remote satellite imaging systems.[1] NOAA subsequently let contracts or bids for the operation of private operation. In 1985 a joint venture of RCA and Hughes won that contract under the assumption that a government subsidy would make up any difference between Landsat's cost and its projected revenues.

The 1984 Act assigned to the National Oceanic and Atmospheric Administration, Department of Commerce, jurisdiction over remote sensing satellites. Title IV of the Act sets up a licensing scene whereby any U.S. firm may apply to own and operate a remote sensing system. As part of the licensing rules, the Commerce Department must consult with the Department of Defense and the Department of State on questions of "national security" and "international obligations."[2]

Any firm that receives a license from NOAA to own or operated a remote satellite carries with it certain obligations and penalties. A violation of a licensing requirement can result in government seizure of satellite images or tapes and the imposition of fines of up to $10,000 per day.

A definitional question persisted, however. What did national security and international obligation entail? What did it mean? Did it imply, for example, that some photos were off limits or that some areas of earth are classified? And did national security argue that some limitation be placed on photo clarity or spatial resolution? And what entities, foreign or domestic, could or might fall within the regulatory scope of NOAA's jurisdiction? To address these questions NOAA instituted a formal inquiry or docket last year.[3] Several participants responded to that docket and in July, 1987, the department, after consulting with the Pentagon and State Department, issued its findings. A first issue turned on the appropriate meter resolution and geographic coverage available to firms engaged in private remote sensing. The Radio and Television News Directors Association (RTNDA) and others insisted that short of an explicit definition of national security, any potential investment in remote sensing stations would be chilled by the uncertainty of those regulations.[4] On the other hand, the Department of Defense stated that the Secretary of Defense should be given:

> "...maximum discretion to determine consistent with applicable First Amendment standards, those conditions necessary to address national security concerns of the United States."[5]

In effect, the Department of Defense requested a veto power over licensees engaged in remote sensing operations. The exact definition of what defense meant by national security, however, has never been spelled out.

A second issue focussed on who qualified for licensing. EOSAT, now operating Landsat 4 and 5, observed that NOAA should license any entity having a "competitive impact" on the commercial satellite sensing market within the United States.[6] That position was predictable. No regulated firm wishes to see its rivals reside outside the burden of licensing requirements of NOAA for fear of placing the former firm at a competitive disadvantage.

The State Department joined EOSAT's position and opted for expanded NOAA. regulation. The Department insisted that no entity, private or public, foreign or domestic, should be excused from government oversight.

Under the State Department's rule, for example, the French SPOT satellite, the German system through its MBB and the European Space Agency would qualify for U.S. licensing once the photos were sold in the Continental U.S.[7] .

In July of this year, NOAA issued its docket recommendations. The final rules became effective on August 10, 1987. The Department insisted that the Department of Defense retain veto power over remote sensing satellites under a national security imperative. On the other hand, NOAA confined its licensing authority to private entities rather than government remote sensing operations. Thus, under NOAA's interpretation, the French SPOT satellite system, although it competes in the United States with U.S. clients, resides outside the direct purview of U.S. regulation.

Clearly, one unanswered question of national security is the issue of spatial resolution or the clarity of remote imaging photos. Over time, technology has generated improvements in photo detail. In the 1960s, for example, images of 1,000 meters were considered state of the art. In the 1970s resolution power was reduced to 100, then 50, and by the 1980's to 30 and 10 meters. Imaging enhancement techniques further improve spatial detail.

But at what point does resolution clarity breach national security standards and interests? A first benchmark occurred during the Carter Administration, according to Robert Aamoth, a Washington attorney. The president issued a classified executive order that placed a 10 meter limit on commercial imaging.[8] Presumably, images of 20 or 30 meters would not compromise national security standards; images of 10, 5 or 3 meters might be declared off-limits by the Department of Defense.

But what standards are to be applied under the Land Remote Sensing Commercialization Act of 1984 remains unknown. It is here that the Secretary of Commerce in consultation with the Secretary of Defense plays a critical role. And it is this unspecified resolution and image availability that raises uncertainty with respect to private imaging prospects.

One such concept, a joint venture of broadcasters or publishing concerns, Mediasat, is exploring the possibility of a 5 meter resolution near real time technology that is permitted by today's off the shelf equipment. Whether such a private imaging system will be sanctioned or approved by the Department of Defense remains very much an open question.

As NOAA regulations now stand, any private entity of the U.S. contemplating a remote satellite sensing system must secure a license from the Department of Commerce. That license is conditioned by the resolution requirement of the Department of Defense and its determination of the availability of those photos to the general public. These standards of resolution presumably will be decided on a case by case basis.

Any prospective firm must submit its spatial resolution standards for DOC/DOD approval, and at any time the Department of Commerce can inspect the premise of the licensee to determine whether compliance of its rules have been violated. In addition, satellite operators must make data available to all buyers on a nondiscriminatory basis. A media company operating a mediasat would be required to sell images to the government.

III. International Entry

In 1986 the French SPOT satellite system broke the U.S. monopoly in commercial remote satellite sensing. Since that time, several nations have announced commercial remote satellite sensing services including the People's Republic of China, Japan, West Germany, Brazil, Canada, India, Indonesia, and Australia. Still more recently, the USSR has offered to sell its remote satellite services on a fee basis through the foreign trade organization Suyuzkarta.

The spatial resolution of many offshore sensing systems exceeds the 30 meter resolution of Landsat. In fact, the French Spot satellite pictures of 10 meter resolution have been displayed on American television. Surprisingly, the Soviet satellite system now offers images of 6 meter resolution, and the Soviet system announced that while rival systems can take pictures of ships, the Soviet image reveals the details of the ship's lifeboats.[9]

Only last June, Canada announced its intention to develop a synthetic aperture radar satellite which can "see through" fog, clouds or the night. West Germany is introducing a 5 meter satellite system by 1990, a so-called modular opto-electronic, multi-spectral scanner. As new entities enter the remote sensing market, these firms will present a new range of options to clients and users, both within and outside the U.S. Over time remote satellite sensing operations will experience intensive price, feature and technological competition.

Competition predictably expedites the rate of technological innovation. Japan, for example, has developed a carbon fiber construction of satellites that permits the lofting of 8 transponders versus the 12 transponder satellite associated with U.S. operations. Thus new technology, new materials, new efficiencies, new productivity as well as new players confront U.S. suppliers of satellite remote sensing systems.

A second area of international rivalry resides in the booster market. The Challenger loss and the delays with expendable launches has created a shortage in the booster market today. Both the French and the People's Republic of China have moved into this market and have gained customers.[10] Today the French Ariene enjoys a backlog of 2.5 billion dollars and is said to to have secured 50% of the world's communication satellite market. The Chinese launching vehicle has both U.S. as well as Sweden and Brazil as potential customers or clients.

Recently, the Soviet Union announced its entry into the commercial launching business at prices at less than half of the U.S. or French boosters. The People's Republic of China has also announced prices of less than 50 - 60% of the Shuttle. Given the shortage of boosters, two firms, General Electric and Hughes have requested permission from the State Department to export satellites to the U.S.S.R. for boosting purposes. The State Department has apparently rejected sending U.S. satellites to the Soviet on grounds that the satellite violates the 1984 International Traffic In Arms regulations.[11] Stated differently, remote satellite sensing falls under the category of munitions according to the State Department.

Competition for satellite boosters derives from firms also in the Far East. The Japanese space program is targetting commercial international customers in the near future. As one official in Japan noted recently:

> "We can accommodate a global market - possibly we can reach to the level of mass launching and mass production satellites."[12]

The fact that a Japanese sensor picked up the X-ray emissions of an exploding star suggests this technology expertise is growing daily. Third World countries moving into satellite boosters include Indonesia, India, and Brazil.

These two forms of entry, remote sensing and launching vehicles, are growing and expanding over time. Competition is thus becoming a process as entry spurs price competition and product innovation particularly in image clarity. The U.S. Geological Survey, the Department of Interior, has even requested permission from U. S. authorities to purchase 6 millimeter images from the Soviet Union. [13]

IV. Policy Issues

The tension between domestic regulation and international competition raises a series of public policy questions. First, do national security regulations inhibit U.S. firms from competing in the domestic market? Assume that the regulation is limited to 10 meters by the Department of Defense. Can an offshore entity enter the U.S. selling 6 meters; and can a U.S. firm compete with its international counterpart for U.S. customers and subscribers? And if NOAA regulates EOSAT or, say, a Mediasat, can these firms compete against the French Spot satellite system exempt from U.S. regulation of spatial imaging? The answer thus far, is that regulation appears to handicap the ability of U.S. firms to compete within the U.S. market.

Second, can U.S. firms compete in the satellite imaging export market? If a ten meter resolution limit is imposed when overseas rivals offer more powerful images, is it not possible that domestic regulation will mean the U.S. will forfeit its ability to compete in the overseas market? EOSAT, for example, operates some 14 receiving dishes throughout the world. They can be modified to receive Spot images from the French. Can these stations remain competitive as the next generation of remote satellites comes on stream with higher resolution? Again, the answer is unknown.

Third, if offshore launching costs represent 60 percent or half of the cost of the U.S. boosters, such as the People's Republic and the USSR, can the U.S. compete internationally on a price basis, assuming an equality of spatial resolution and photo availability. And should any U.S. client possess the ability to pick low cost, irrespective of source, in order to retain customers, and remain competitively viable.

Fourth, U.S. firms supply components to offshore remote satellite systems. If these systems are sent to third parties, for example South Africa, will the US Department of Commerce or State or Defense invoke national security laws, export laws, to deny the components for these systems? While effective in the short run, will not such regulations mean that international firms will move into the component market and thus represent a loss of business to U.S. firms over the long haul. And what does regulation do to the U.S. trade deficit?

Finally, will the Department of Defense or State find it necessary to augment the integrity of domestic regulation and extend national defense jurisdiction to NATO and non-NATO firms - a form of the COCOM laws applied beyond the jurisdictional reach of the U.S. That is, once regulation commences and is applied to U.S. firms is it not inevitable they be extended to the competitors of U.S. firms in order to reach equality of national security definition.

Conclusion

The U.S. transition from a government to a private operation of remote satellite systems is not without trauma or difficulty.

In the process, private remote satellite systems are caught between domestic regulation and global competition. On the domestic side, the shuttle is grounded until 1988, and when available, military satellites will obviously assume first priority. Expendable launching vehicles are slowly coming onstream, the life of the Landsat 4 and 5 is about to expire, and EOSAT's financial condition remains tenuous. Private entities are beset by uncertainty and risk as to regulation of photo availability and spatial resolution.

In the meantime, consumer options in satellites and boosters are expanding inexorably - beyond the U.S., beyond Europe, beyond the Soviet Union, to Third World countries. Regulations that determine spatial resolution and geographic availability, however valid in the past, are likely to be circumvented as more and more rivals introduce subsequent generations of remote satellite systems of lower cost and increased power.

All of this suggests that foreign competition holds the potential of eroding the U.S. definition of national security. Here the U.S. will be caught in a painful dilemma - namely, to relax national security standards to enable U.S. firms to compete internationally, or extend those regulations to overseas rivals as a means of levelling the rules of the game. Whatever the case, the choice will not be easy, for it is clear that international rivalry is erasing clear distinctions between commercial and military products, service and technology.

Footnotes

1. 15 U.S.C., p. 4201-4292.

2. Ibid, p. 4201

3. Department of Commerce, In the Matter of the Licensing of Private Remote Sensing Space Systems, National Oceanic and Atmospheric Administration, Docket No. 51191-5191, Notice of Proposed Rulemaking, March 24, 1986. (Cited as NOAA Docket No. 51191-5191).

4. Docket No. 51191-5192, Comments of Radio-Television News Directors' Association, American Society of Newspaper Editors, National Association of Broadcasters, National Broadcasting Company, Inc., Society of Professional Journalists, Sigma Delta Chi and Turner Broadcasting System, Inc., February 11, 1987.

5. N.O.A.A. Docket No. 51191-5191, Department of Defense, December 31, 1986, p. 2.

6. N.O.A.A. Docket No. 51191-5191, Final Rule, July 16, 1987, p. 7

7. N.O.A.A. Docket, State Department Comments, December 29, 1986.

8. Robert A. Aamoth, "From Landsat to Mediasat: The Development of Remote-Sensing Technology and the First Amendment Right of the Press to Use that Technology for News Reporting," American Enterprise, The Law and The Commercial Use of Space, Vol. II, National Legal Center for the Public Interest, Washington, D. C., 1986.

9. "Soviets Offer High-Resolution Sensing Data," Space Business News, Pasha Publications, July 10, 1987, p. 2.

10. Craig Covault "Chinese Expand Launch Facilities to Attract Satellite Customers," Aviation Week and Space Technology, July 13, 1987, p. 120; also, Less Dorr, " Russki Business" Space World, July 1987, p. 32.

11. "State Dept. Denies License to Export U.S. Satellites to the Soviet Union," Aviation Week and Space Technology, July 27, 1987, p. 59; See also Theresa M. Foley, "Satellite Builders Want Change in U.S. Anti Proton Policy," Aviation Week and Space Tehcnology, September 28, 1987, p. 138.

12. Tom Ashbrook, "Japan Intent on Making Its Own Way Into Space," Boston Globe, February 23, 1987, p. 34.

13. William Broad, "Sales Blitz by Soviet Seeks to Tap Global Market in Space Services," New York Times, September 6, 1987, p. 1; Also W. Broad, "Soviet Space Photography Stirs U.S. But Agency Plan to Buy Them Faces Barriers," New York Times, September 27, 1987, p. 40.

STORAGE AND RETRIEVAL

BLANK PAGE

Multi-Colored Quadtrees for GIS: Exploiting Bit-Parallelism for Rapid Boolean Overlay

Sudhakar Menon
Peng Gao
Terence R. Smith
The University of California, Santa Barbara, CA 93106

Abstract

At the present time a thematic layer is usually represented in quadtree based GIS using a collection of binary quadtrees, one for each color or value in the layer. In such a scheme set operations are carried out by taking unions and intersections of binary quadtrees. This paper describes an alternative database scheme where a layer is stored as a single multi-colored quadtree, using a bit-list to store values at higher nodes in the tree. This allows the union of any set of values within a layer to be carried out by a single top down traversal of the tree, using bit-parallel logical operations. Recent developments in quadtrees have led to space efficient hierarchical representations one of which is the Autumnal quadtree. Experimental results are presented which show that representing a layer using a Bit-Mapped Multi-Colored Autumnal quadtree is more space efficient in comparison to using of a collection of Binary Autumnal quadtrees, one for each value in the layer. Evidence is also presented to show that these Bit-Mapped Multi-Colored quadtrees lead to faster color selection and boolean overlay between layers in a GIS as compared to performing set operations on Binary quadtrees.

1. Introduction

A number of tesselation based Geographic Information Systems using hierarchical data structures based on the quadtree model are currently in existence. Hierarchical data structures have gained acceptance because they potentially offer reduced space requirements and faster algorithm execution times as compared to non-hierarchical structures.

Region Data form an important constituent of any spatial database. The most frequently used operation on such data involves the spatial overlaying of different thematic layers in the database. Each thematic layer may be viewed as a partition of space into different regions, each with a unique value. The overlay operation may be viewed as the search for those regions of space that correspond to a specified union of values within each layer. An example of such a query is the search for all regions where :

[LANDUSE = (10, 11, 15)] AND [ELEVATION = (200 300)]

A set of discrete values is specified for Landuse. Elevation is continuosly valued, and is specified using a lower and upper limit to indicate the desired range of values. The regions returned in answer to the query must simultaneously possess acceptable values in both the layers.

In designing databases using the quadtree model for geographic applications a key design decision involves the choice between binary and multi-colored quadtrees. A thematic layer with *m* values, or colors, may be represented using a set of *m* binary quadtrees, one for each color. Alternatively, a layer may be stored in a single multi-colored quadtree, with *m* colors. At the present time the choice seems to be in favor of representing each layer as a set of binary quadtrees. Overlay operations are then formulated in terms of intersections and unions between binary quadtrees. In this paper we introduce Bit-Mapping, a new scheme for storing thematic information in multi-colored quadtrees. The scheme provides for efficient information storage while simultaneously allowing the bit parallel capabilities of sequential machines to be used in overlay operations. By formulating the union operation in terms of a top down search of the multi-colored quadtree, the time complexity of overlay operations is improved. An analysis together with experimental results comparing the two alternative schemes for storing multiple valued thematic data is presented.

2. Quadtree Data Structures

A quadtree is used to represent the process by which a $2^n \times 2^n$ binary array may be recursively subdivided into quadrants or blocks, till each block contains a uniform value. The tree has a branching factor of 4, the root node corresponds to the entire array and the four sons of a node correspond to the quadrants into which it is divided. Leaf nodes correspond to the largest quadrants with a uniform value, and may occur at any level in the tree.

Regular Quadtrees The regular quadtree also referred to as the pointer based quadtree corresponds to a direct implementation of the above description using a pointer based structure. Each node of the tree is stored as a record with six fields (Samet, 1980). Four fields contain pointers to sons, a fifth field contains a pointer to the father, and a sixth field contains the nodes value or color (GRAY for interior nodes, BLACK or WHITE for leaf nodes, for a binary array). The main disadvantage of the structure is the high internal storage cost associated with the need to store 5 pointers with each node.

Key Based Quadtrees Key based structures represent the quadtree as a list of leaf nodes using a linear key that encodes the depth and spatial location of leaf nodes. The Linear quadtree, introduced by Gargantini (1982) uses a locational code to represent each BLACK leaf node in the tree. Each node is encoded using a ternary integer whose digits reflect successive quadrant subdivisions. The Linear quadtree for a binary image consists of a sorted list of the locational codes of BLACK leaves, corresponding to the order in which the leaves are encountered in a postorder traversal of the corresponding regular quadtree. A fixed length key similar to Gargantini's was also introduced by Abel and Smith (1983). If *N* is the total number of nodes in a regular quadtree then the best case and worst case number of nodes in the corresponding Linear quadtree are 0.17*N* (one BLACK node for every three WHITE nodes) and 0.54*N* (three BLACK nodes for every WHITE node) respectively. The main disadvantage of Key based quadtrees is that the absence of internal nodes does not permit the storage of a hierarchical view of the data.

Autumnal Quadtrees The Autumnal Quadtree is a hierarchical structure that retains the advantages of the Regular quadtree but achieves space savings by eliminating the need to store leaves (Fabrini and Montani, 86). The relevant information for each leaf node is codified into its fathers pointer field. Each node in the tree is a GRAY node and the color or value field can thus be removed. If N is the number of nodes in a Regular quadtree, the corresponding Autumnal Quadtree has only $0.25N$ nodes, corresponding to the GRAY nodes in the Regular Quadtree. Using Autumnal Quadtrees therefore results in a space savings of 75%.

Multi-Color Extensions Both the Regular and the Autumnal Quadtree are easily extended from the binary to the Multi-Colored case. The value field in an internal node of a Regular quadtree may be used to store an appropriate function of the values in the corresponding portion of the array. Each leaf node represents a quadrant with a uniform color, which is stored in its value field. Linear quadtrees are by definition lists of BLACK nodes, and contain no representation for internal nodes. They are thus not extendable to the multi-colored case. A multiple values array must therefore be represented using a set of binary Linear quadtrees, one for each value.

3. Quadtree based GIS

A number of GIS using quadtrees are described in the literature. At present the Linear quadtree is the most widely used data structure. Irrespective of the underlying quadtree data structure, a multiple valued thematic map is usually represented using a set of binary quadtrees, one for each distinct value. Samet et. al. (87) describe the development of a quadtree based GIS at the University of Maryland. While the initial implementation (Rosenfeld et. al. 82) was pointer based, later implementations used the Linear quadtree data structure primarily because of the reduced storage requirements of the latter. The availability of efficient schemes for external storage management has also been a factor in the widespread use of Linear quadtrees. For example, Abel (1984) discusses a B+ tree organization for managing large linear quadtrees on external storage based on a locational key. Several efficient algorithms eliminating the need for neighbor searching in operations such as connected component labeling (Samet and Tamminen, 85) have also improved the time performance of linear quadtree based systems.

The recent development of space efficient storage schemes has made hierarchical tree structures attractive. Fabrini and Montani (86) describe a geographic database built using Autumnal quadtrees. The database represents a multiple valued thematic map using a set of N binary Autumnal quadtrees. Peng and Smith (87, this volume) describe a database and an efficient external storage management scheme using Relatively Addressed Compact quadtrees, which combine elements from the One-to-Four and Autumnal data structures with additional value coding, to yield structures that compare with linear quadtrees in storage requirements. This paper examines the issue of Binary versus Multi-Colored quadtrees in the light of these recent developments.

4. Bit-Mapping for Multi-Colored Quadtrees

Given a hierarchical structure built on a multiple valued array, a number of schemes may be employed to store information in the internal nodes of the tree. The value field of an internal node of a Regular or an Autumnal quadtree may thus store the average, the minimum, the maximum, or any other appropriate function of the values in the portion of the array covered by the node. Alternatively, the complete set of values present in the portion of the array covered by the node may be represented.

Bit-Mapping refers to a compact way of storing the complete set of values covered by a node using a fixed length bit-list. One bit is allocated for each possible value in the underlying array. An m valued array thus requires each internal node to have a value field that is m bits long. A bit in the bit-list is set if and only if it corresponds to a color that is present in the portion of the array covered by the node. In our GIS implementation, we use a bit-list of length 32, corresponding to the word size of many available computers. This allows a thematic layer to contain up to 32 discrete values, adequate for most purposes.

5. Bit-Parallel Searching

The process of extracting the binary quadtree corresponding to the union of a set of values in the underlying array is carried out by bit-parallel top down searching. The extraction process utilizes the parallel bit-handling capabilities of most sequential computers, to classify an internal node of the multi-colored quadtree as BLACK, WHITE or GRAY with respect to the binary quadtree that is being extracted.

In order to extract the binary quadtree representing the union of k values in the underlying array, the set of desired values is itself represented using a bit-list, referred to as the **template_list**. Let **value_list** represent the bit-list storing the actual values covered by an internal node. Let **empty_list** represent the empty bit-list (corresponding to decimal 0 represented using 32 bits). It is assumed that neither **value_list** or **template_list** is equal to **empty_list**. An internal node of the multi-colored quadtree is classified with respect to the binary quadtree being extracted using the rules described below. The description of each rule is followed by the bit-parallel expression used to evaluate its truth condition.

BLACK : A node of the multi-colored quadtree is BLACK with respect to the binary quadtree representing the union if and only if every value covered by the node is a member of the set of sought for values. The node is thus BLACK iff
(template_list BIT-OR **value_list)** EQUALS **template_list**

WHITE : A node of the multi-colored quadtree is WHITE with respect to the binary quadtree representing the union if and only if no value covered by the node is a member of the set of sought for values. The node is thus WHITE iff
(template_list BIT-AND **value_list)** EQUALS **empty_list**

GRAY : A node of the multi-colored quadtree is GRAY with respect to the binary quadtree representing the union if and only if it is neither BLACK nor WHITE.

BEST COPY AVAILABLE

The extraction of the binary quadtree corresponding to the sought for union is accomplished using a simple preorder traversal of the multi-colored quadtree, beginning at the root, or at the node covering the search window. Each node is classified as BLACK, WHITE or GRAY. If a node is GRAY its children are recursively examined. The classification rules described above are implemented using the bit-parallel logical operations BIT-OR and BIT-AND.

6. Space Requirements

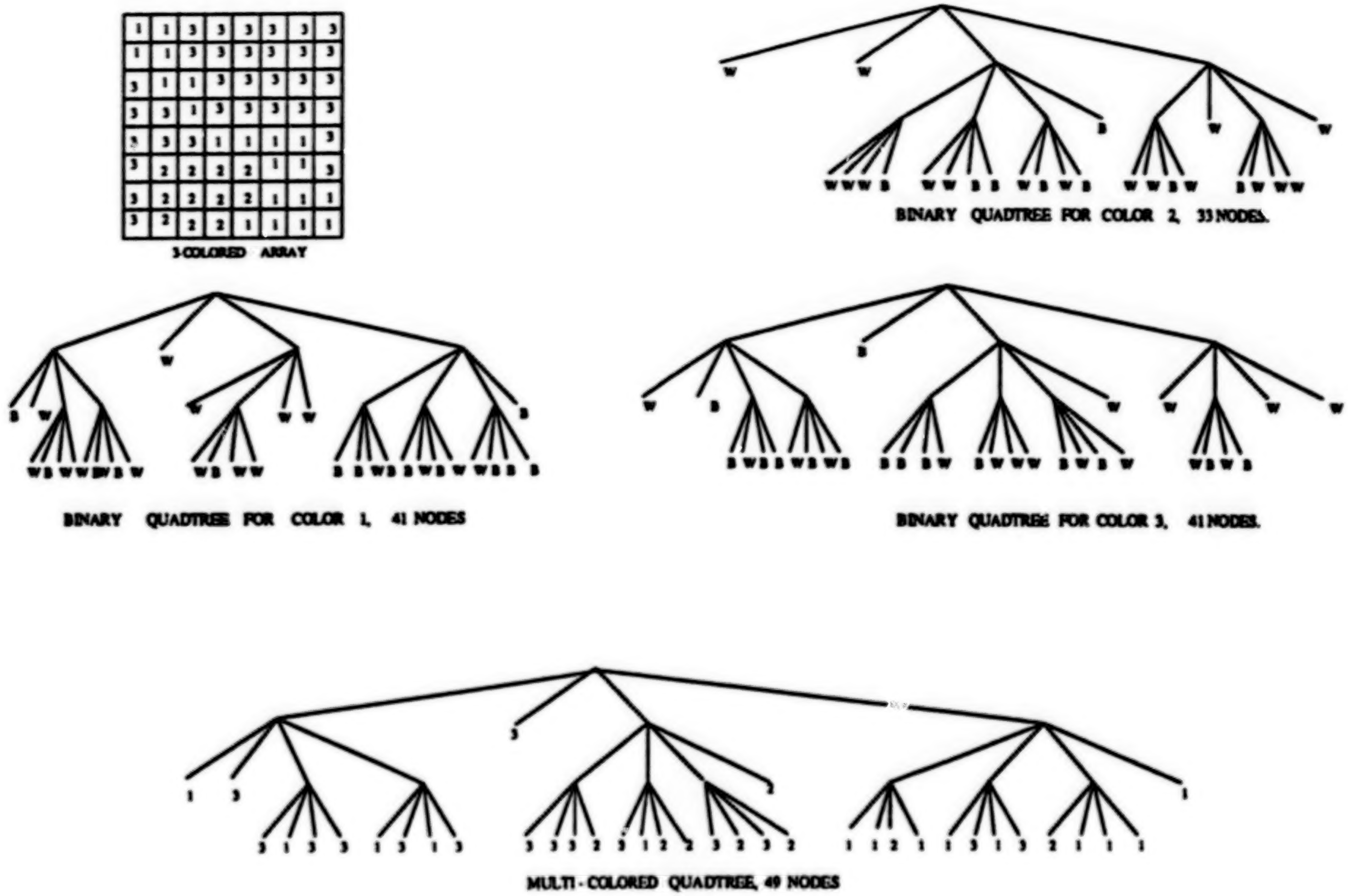
In this section we compare the cost of storing a multiple valued thematic layer as a set of Binary quadtrees with the cost of storing it as a Bit-Mapped Multi-Colored quadtree. Consider a thematic layer with m distinct values. Let N_i represent the number of nodes in the Regular, Binary quadtree for value i. Let N' represent the number of nodes in the Regular, Multi-Colored quadtree corresponding to the array. The total number of nodes required for the two schemes is then as shown in Table 1, below :

Table 1 : Number of Nodes Compared

Data Structure	Total Binary Set	Bit-Mapped Multi-Colored tree
Regular	$\sum_{i=1}^{i=m} N_i$	N'
Autumnal	$\sum_{i=1}^{i=m} 0.25\times N_i$	$0.25\times N'$
Linear	$\sum_{i=1}^{i=m} (0.17-0.54)\times N_i$	-

The relation between the total number of nodes in the set of Regular Binary quadtrees and the number of nodes in the Regular Multi-colored quadtree is illustrated using the 3 colored example array of Fig. 1. The figure shows a three valued array, the three Regular Binary quadtrees and the Regular Multi-Colored quadtree. For this array, $N_1 = 41$, $N_2 = 33$, $N_3 = 41$ and $N' = 41$. The number of nodes in the Multi-Colored Regular quadtree is less than half the total number of nodes used in the set of Binary Regular quadtrees.

Data on the total number of nodes used by the two schemes were generated for the Landuse and Geology layers of a database for the Black Hills Region, S. Dakota, USA. The quadtrees were built from multiple valued raster datasets, 512 x 512 pixels in size. The data generated for Landuse (22 discrete values) and for Geology (31 discrete values) are shown in Table 2, below. The table also shows the actual space used, in 512 byte blocks, for the set of Autumnal Binary quadtrees and for the corresponding Bit-Mapped Multi-Colored quadtree.



1	1	3	3	3	3	3	3
1	1	3	3	3	3	3	3
3	1	1	3	3	3	3	3
3	3	1	3	3	3	3	3
3	3	3	1	1	1	1	3
3	2	2	2	2	1	1	3
3	2	2	2	2	1	1	1
3	2	2	2	1	1	1	1

3-COLORED ARRAY

BINARY QUADTREE FOR COLOR 2, 33 NODES.

BINARY QUADTREE FOR COLOR 1, 41 NODES

BINARY QUADTREE FOR COLOR 3, 41 NODES.

MULTI-COLORED QUADTREE, 49 NODES

FIGURE 1 : NODE SIZES COMPARED

Table 2 : Data Storage Requirements Compared

Layer	Description	No: of Nodes		Autumnal
		Linear	Autumnal	File Size
Landuse	Total Binary Set	19336	13670	364
Landuse	Multi-Colored Tree	-	6434	180
Geology	Total Binary Set	69274	52075	1143
Geology	Multi-Colored Tree	-	23070	576

For each layer the table shows the total number of nodes required for the complete set of Binary quadtrees, using the Linear and Autumnal data structures. The table also shows the number of nodes in the Bit-Mapped Multi-Colored quadtree for the layer, using the Autumnal data structure. For both layers, the number of nodes required by the Bit-Mapped Multi-Colored Autumnal quadtree is seen to be slightly less than **half** that required by the total set of Binary Autumnal quadtrees, and approximately **one-third** that required by the total set of Binary Linear quadtrees.

A comparison of the actual space requirements of the two schemes was carried out using an Autumnal quadtree based implementation. An interior Autumnal quadtree node stores 4 pointers to its sons. A pointer to the father is not stored as top-down algorithms do not require it. An interior node of a Binary Autumnal quadtree requires no value field and is hence implemented using 8 bytes. Since the Autumnal quadtree has one-fourth the number of nodes of the Regular quadtree, the cost per node of the corresponding Binary Regular quadtree is therefore 2 bytes. A node of the Bit-Mapped, Multi-Colored Autumnal quadtree is implemented using 4 bytes to store the bit-list of length 32 at each internal node and 8 bytes for four pointers. When leaf values are pulled up into the pointer space of the parent, they are stored as actual values (not bit-lists) in the parents pointer field. The total cost of a node is thus 12 bytes. The cost per node of the corresponding Multi-colored Regular quadtree is therefore 3 bytes. A Linear quadtree node requires 4 bytes of space in most reported implementations. For the two layers studied the fraction of BLACK leaves in the Regular quadtree is approximately 0.33 (as compared to a best case ratio of 0.17 and a worst case ratio of 0.54). The cost per node of a Binary Regular quadtree is therefore 1.32 bytes.

Taking relative node sizes into account, the expected space savings, neglecting overheads, when using a Bit-Mapped Multi-Colored Autumnal quadtree as compared to a set of Binary Autumnal quadtrees is 25 %. Taking relative node sizes into account the expected space requirements, neglecting overheads, of the Bit-Mapped Multi-Colored Autumnal quadtree are the same as that for the set of Binary Linear quadtrees. The table also compares the actual file storage for these two alternative schemes in units of 512 byte blocks, for each layer, using the Autumnal quadtree data structure. For each layer, the Bit Mapped Multi-Colored Autumnal quadtree is shown to yield a space saving of approximately 50 %. This is much better than the expected value because of the reduced file management overhead when using the Multi-Colored quadtrees. Storing an m valued layer using a set of Binary quadtrees requires the use of m files, one for each value. Storing the layer using a Multi-Colored quadtree however requires only one file. Managing quadtrees in external storage for large databases invariably requires efficient spatial indexing, with the attendant storage overhead. The external storage management scheme used in our implementation is described elsewhere (Peng and Smith, 87). Each 512 x 512 image incurs an overhead of 2 blocks in storing the index. Additionally, the total overhead caused by unused but allocated fractional blocks is greater when using m files. Taking these factors into account, the actual savings increase to approximately 50 %. An identical argument for file overhead applies to the use of a set of Binary Linear quadtrees. In terms of an actual implementation, we may therefore expect the Bit-Mapped Multi-Colored quadtree to be more space efficient. Finally, it must be reiterated that the Bit-Mapped Multi-Colored quadtree provides hierarchical structure not present in the set of Binary Linear quadtrees : An overview of the data values present in any internal quadrant or block of the layer is always immediately available.

7. Time Complexity

In this section we compare the time complexity of extracting the Binary quadtree corresponding to the union of k distinct values using the two alternative schemes. When using a set of Binary quadtrees, algorithms for quadtree union are employed. When using Bit-Mapped Multi-Colored quadtrees, top down search is used. Costs are compared using the number of quadtree nodes examined in each case.

7.1. Union of a set of Binary Autumnal quadtrees

Let N_i be the number of nodes in the Regular Binary quadtree for value i. Let N_r be the number of nodes in the Regular Binary quadtree that represents the resulting union. If Regular or Autumnal quadtrees are used, the union of the k values is accomplished by applying a pairwise tree traversal algorithm (Hunter and Steiglitz, 1978). After the union of the k trees has been taken an additional traversal of the resulting Binary quadtree is required for condensation. In the best case the resulting tree has exactly N_r nodes, and no actual condensation takes place. A worst case bound on the number of nodes in the resulting quadtree after union and before condensation is the sum of the nodes in the input binary quadtrees. The total cost of the union is thus bounded between :

$C_1.\sum_{i=1}^{i=k} N_i + C_2.N_r$ and $(C_1+C_2).\sum_{i=1}^{i=k} N_i$, where C_1 and C_2 are constants.

7.2. Union of a set of Binary Linear quadtrees

Let n_i be the number of nodes in the Linear Binary quadtree for value i (Note : $n_i=(0.17-0.54).N_i$). Let n_r be the number of nodes in the Linear Binary quadtree representing the resulting union. The cost of taking the union of the k values is proportional to the sum of the nodes in the k Binary Linear quadtrees (Bauer, 85, Abel, 85). Again condensation is required to ensure that the resulting quadtree has maximal blocks. The total cost of the union is then bounded between :

$c_1.\sum_{i=1}^{i=k} n_i + c_2.n_r$ and $(c_1+c_2).\sum_{i=1}^{i=k} n_i$, where c_1 and c_2 are constants.

7.3. Top Down Search of the Bit-Mapped Multi-Colored quadtree.

The Binary quadtree corresponding to the union of k distinct values is extracted from the Bit-Mapped Multi-Colored Autumnal or Regular quadtree with m values using a top down pre-order traversal. Each node examined is classified as BLACK, WHITE or GRAY with respect to the Binary quadtree being extracted. The children of GRAY nodes are recursively examined. Examination of each node takes a *constant* amount of time, because of the bit-parallel operations used in classification. The total time spent in search is therefore proportional to N_r, the number of nodes in the resulting Binary, Regular quadtree corresponding to the union of the k specified values. Since the search proceeds top-down each node classified is a maximal block and children of BLACK and WHITE nodes are never examined. The process thus requires **no condensation.**

The cost of top-down search is thus $C_3 N_r$, where C_3 is a constant.

7.4. Experimental Comparison

Experiments were run in order to compare the time requirements of the two alternative algorithms using four sets of desired values, from the Landuse and Geology layers. Table 3. shows data on the number of nodes using the notation discussed in sections 7.1, 7.2 and 7.3.

Table 3 : Data on Number of Nodes

Layer	Values in Union	$\sum_{i=1}^{i=k} N_i$	N_r	$\sum_{i=1}^{i=k} n_i$	n_r
Landuse	6,11,13,14	36644	12764	13792	5374
Landuse	1,2,6,9,12,13	36468	15764	13678	6743
Geology	5,9,12,13	55952	51460	19414	18214
Geology	5,9,12,13,23,24	75112	50332	25958	18614

Table 3. shows that the relationship between N_r and $\sum N_i$ varies with the geometric distribution of the data and the resultant amount of condensation required when forming the quadtree representing the union. The sum of the nodes in the set of Binary Linear quadtrees, $\sum n_i$, is between 34 % and 37 % of $\sum N_i$ for all four cases.

Table 4 compares algorithm costs, estimated in terms of the number of nodes. Condensation costs for the union algorithms are estimated by adding the number of nodes in the resultant quadtree to the number of nodes in the input binary quadtrees. The comparison thus assumes a value of 1 for the constants C_1, C_2, c_1, c_2 and C_3 described in sections 7.1, 7.2 and 7.3, and uses the lower bound for union costs. For top-down search of the Bit-Mapped, Multi-Colored, Autumnal quadtree, the actual number of nodes, N_r, is used. The data in Table 4 serve as an approximate index of comparison, as all constants are assumed equal, and the cost of condensation is only estimated.

Table 4 : Algorithm Costs in Number of Nodes

		Binary Union		Top-Down Search
Layer	Values in Union	Autumnal	Linear	Autumnal
Landuse	6,11,13,14	49408	19166	12764
Landuse	1,2,6,9,12,13	52232	20423	15764
Geology	5,9,12,13	107412	37628	51460
Geology	5,9,12,13,23,24	125444	44572	50332

The table shows that the number of nodes visited in searching the Bit-Mapped Multi-Colored Autumnal quadtree is smaller than the number of nodes examined in taking the union of the corresponding set of Binary Autumnal quadtrees in all cases. The number of nodes visited in searching the Bit-Mapped Multi-Colored Autumnal quadtree is, on average, of approximately the same size as the number of nodes examined in taking the union of the corresponding set of Binary Linear quadrees. This comparison is not directly translatable into execution times as the different algorithms involve different constants of proportionality, and the cost of condensation varies with the union algorithm used.

Experiments were carried out to compare the actual running times of the top-down search algorithm for the Bit-Mapped Multi-Colored quadtree with the Union algorithm for the corresponding set of Binary Autumnal quadtrees, using the memory management system described in Peng and Smith (87). Table 5 compares execution times in CPU seconds on a VAX 11/750.

Table 5 : Execution Times using Autumnal Structures

Layer	Values in Union	Binary Union	Top-Down Search
Landuse	6,11,13,14	8.17	5.77
Landuse	1,2,6,9,12,13	9.15	6.11
Geology	5,9,12,13	14.59	12.09
Geology	5,9,12,13,23,24	21.92	12.28

In all cases, the top-down search of the Bit-Mapped Multi-Colored quadtree is seen to be faster. No actual running time comparison is available with the corresponding union operations when using Binary Linear Quadtrees with a disk based memory management system.

8. Continuous Data

The preceding discussions have focused on the use of discrete layers with nominal values such as Landuse and Geology. The multi-colored quadtree approach described extends directly to continuous data such as Elevation and Slope. Since the data is continuous, the complete set of values covered by a node is adequately represented by storing the minimum and the maximum values covered. Additional data, such as the mean value, may also be stored.

The analog of the union operation is the extraction of all regions corresponding to a specified range of values. Extraction may again be carried out using top-down search, classifying each node of the multi-colored quadtree with respect to the binary quadtree that is being extracted. The desired range of values, referred to as the **template-range** is specified using a minimum and a maximum value. This range is compared with **value-range**, the actual contents of the node. A node is classified as WHITE if the two ranges are completely disjoint. If the **value_range** is completely contained within the **template_range** the node is classified as BLACK. Else, the ranges overlap, and the node is GREY. Given a **continuous** range of sought for values, the classification of a node is therefore seen to take constant time.

The space and time comparisons between Bit-Mapped Multi-Colored quadtrees and the equivalent set of Binary quadtrees extend to Multi-Colored quadtrees for continuos data. Additionally a finer level of data quantization is usable with the multicolored scheme. Each node of an Autumnal Multi-colored quadtree for continuos data can be implemented in 12 bytes, of which 4 bytes can be allocated to storing the minimum, maximum, and optionally the mean values. The cost per node of the Regular quadtree is thus still 3 bytes, as for the Bit-Mapped case. Finally we note that the use of the Compact Quadtree scheme for multi-colored continuos data with two levels of coding, described in Peng and Smith (87), further reduces storage costs to approximately 1.5 bytes per node of the corresponding Regular quadtree. Under these conditions the Multi-Colored quadtree for Continuos data will be more storage efficient than even the set of Binary Linear quadtrees representation.

9. Conclusions

In this paper we have introduced Bit-Mapping, a technique for the efficient storage of information in the interior nodes of multi-colored quadtrees for discrete data. The space and time complexity of Bit-Mapped Multi-Colored quadtrees were analyzed and compared with the complexities for the alternative storage scheme : representing a multiple valued thematic layer using a set of Binary quadtrees.

Empirical investigations of performance were made using the Landuse and Geology layers of a spatial database for the Black Hills, S.Dakota. Representing a layer using a

Bit-Mapped Multi-Colored Autumnal quadtree was found to require half the number of nodes used by the corresponding set of Binary Autumnal quadtrees and one-third the number of nodes used by the corresponding set of Binary Linear quadtrees. Taking relative node sizes into account, the Bit-Mapped Multi-Colored Autumnal quadtree was expected to require 25% less space than the corresponding set of Binary Autumnal quadtrees and approximately the same space as the corresponding set of Binary Linear quadtrees. Actual implemented space savings obtained using Bit-Mapped Multi-Colored Autumnal quadtrees in comparison to a set of Binary Autumnal quadtrees was 50%. The increased space savings resulted from the overhead involved in mantaining a separate indexed file for each color, when using Binary Autumnal quadtrees. A similar indexing overhead may be expected in the mantainence of a set of Binary Linear quadtrees on external storage.

A top down search algorithm for extracting the binary quadtree corresponding to a union of values from the Bit-Mapped Multi-Colored quadtree was described. The classification of a node of the Multi-Colored quadtree as BLACK, WHITE or GRAY with respect to the Binary quadtree being extracted was shown to take constant time, independent of the number of values in the union. The performance of the top down search algorithm was compared to the alternative : taking the Union of the set of Binary quadtrees corresponding to the required values. The top down search algorithm visits fewer nodes than are examined in taking the union of the corresponding set of Binary Autumnal quadtrees. This difference in the number of nodes increases as the amount of condensation required after taking the union increases. Actual time comparisons using Autumnal quadtrees showed the top-down search algorithms to be approximately twice as fast in the majority of cases. The estimated cost, including condensation, of taking the union of a set of Binary Linear quadtrees was found to be approximately equal to that of top-down search of a Multi-Colored Autumnal quadtree. The conclusions drawn with respect to Bit-Mapped Multi-Colored Autumnal quadtrees for discrete data were shown to also apply to Multi-Colored quadtrees for continuos data storing minimum and maximum values.

The Bit-Mapped Multi-Colored Autumnal quadtree representation for a thematic layer is superior to the representation using a set of Binary Autumnal quadtrees, with respect to space and the time required for Union operations. This study indicates that the representation of a layer using a Bit-Mapped Multi-Colored Autumnal quadtree requires no more space and approximately the same time for Union operations as the representation using a set of Binary Linear quadtrees. In addition, Multi-Colored quadtrees easily support the inverse of the query posed in the introduction : Given a region in space, represented as a set of quadtree nodes, return the set of thematic values in any specified layer, that occur within the region. An example of such a query is the production of a table showing Landuse values within a specified region. For any one node in the region, determining the set of Landuse values covered by the node requires *m* binary searches when using a set of *m* Binary Linear quadtrees to represent the Landuse layer as compared to the examination of a single node when using Multi-Colored quadtrees.

References

ABEL, D.J., 1984, A B+ tree structure for large quadtrees, Computer Vision Graphics and Image Processing, 27, 1, pp 19-31.

ABEL, D.J., 1985, Some Elemental Operations on Linear Quadtrees for Geographic Information Systems, The Computer Journal, 28, 1, pp 73-77.

ABEL, D.J. & SMITH, J.L., 1983, A data structure and algorithm based on a linear key for a rectangle retreival problem, Computer Vision, Graphics and Image Processing, 24, 1, pp 1-13.

BAUER, M.A., 1985, Set Operations on Linear Quadtrees, Computer Vision, Graphics and Image Processing, 29, pp 248-258.

FABRINI, F. & MONTANI, C., 1986, Autumnal Quadtrees, The Computer Journal, 29, 5, pp 472-474.

GARGANTINI, I., 1982, An effective way to represent quadtrees, Communications of the ACM, 25, 12, pp 905-910.

HUNTER, G.M. & STEIGLITZ, K., 1979, Operations on Images Using Quad Trees, IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-1, 2, pp 145-153.

PENG, G. & SMITH, T.R., 1987, Space Efficient Hierarchical Structures, Relatively Addressed Compact Quadtrees for GIS, International Geographic Information Systems Symposium, Arlington, Virginia, USA.

ROSENFELD, A., SAMET, H., SHAFFER, C.A. & WEBBER, R.E., 1982, Application of hierarchical data structures to Geographic Information Systems, Computer Science Technical Report TR-1197, University of Maryland, USA.

SAMET, H., 1980, Region Representation : Quadtrees from Binary Arrays, Computer Graphics and Image Processing, 13, 1, pp 88-93.

SAMET, H., SHAFFER, C.A., NELSON, R.C., HUANG, Y.G., FUJIMARA, K & ROSENFELD, A., 1987, Recent developments in linear quadtree-based Geographic Information Systems, Image and Vision Computing, 15, 3, 187-197.

SAMET, H. & TAMMINEN, M., 1985, Computing Geometric Properties of Images represented by Linear Quadtrees, IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-7, 2

BLANK PAGE

PLANNING FOR FUTURE ARCHITECTURE CHANGES

Thomas M. Myers
Arthur Andersen & Co.
101 North First Avenue #2100
Phoenix, Arizona 85003

ABSTRACT

This paper addresses many of the issues involved when transferring geographic information to a different system architecture. A future change in architecture may be necessary for a variety of reasons, ranging from the inadequacy of the current architecture to imposed hardware requirements. Factors which affect the transfer of data include the structure in which data elements are stored, the data relationships, and the functional approach of the software. Various methods can be used to structure data in a format which will facilitate a transfer to a new environment. An analysis of these issues when developing data conversion specifications may result in substantial savings if the hardware or software direction changes in the future. Given the volatile nature of the industry, planning for future architecture changes is essential. This paper also discusses case studies of two organizations which undertook architecture changes. The different backgrounds, objectives, approaches and results of these endeavors will be reviewed.

1. INTRODUCTION

The development of a Geographic Information System (GIS) represents a substantial investment of both time and money. Since the GIS industry is still in its evolutionary stages, both hardware and software are constantly changing. New developments are occurring each year, and vendors are competing to provide the best solution. Many factors may result in a change to the system architecture. The originally selected architecture may not be capable of performing new functions, a new vendor may release a superior system on different hardware, data requirements may outgrow current hardware capacity, or other systems within the organization may dictate a change to a consistent architecture due to interface requirements.

An important issue concerns the nature of an architecture change. A decision to change architectures dictates a complete transfer to a new system. This change is a one-time effort which should not be repeated on a short term basis. The new architecture will replace the existing system. Many organizations are assessing the possibility of operating a system concurrently on two architectures. For example, they seek to maintain a data base on one architecture and graphics software on another. This is an extremely high risk configuration. Current GIS software has integrated the data and graphic components to the extent that separating them would be detrimental to the system. The only way to tie separate systems together is to build a custom bridge between the two architectures and reformat data being passed back and forth. This bridge would be susceptible to any change in either architecture. Given the volatile nature of the software, the interface would be in a constant state of revision, and is liable to be made obsolete at any time.

Regardless of the architecture used, the foundation of the system is the data, which remains relatively stable throughout the life of an organization. The conversion effort for capturing this data represents a major effort (and expense) of system development, and can exceed the cost of hardware and software combined. If the data can be captured in a manner which will facilitate transfer to a future system architecture, then the data conversion costs can be considered an investment, and the magnitude of re-conversion costs can be minimized. Given the current state of the industry, this careful planning for future change is a necessity.

2. CURRENT DATA ELEMENTS

Vector storage has become the standard for many current GIS architectures. Vector elements are much easier to attach to an intelligent infrastructure than their raster equivalents. Vector elements are also much easier to store and retrieve, thus reducing input/output times and improving system response. Conversion processes may capture images in raster form, but these images are converted to vector format before becoming functional in a geographic system. The transfer of data between systems therefore entails the translation of vector files from one format to another.

Vector graphic elements are stored in a relatively consistent manner across different architectures: a line consists of two end points; a circle is a center point and a radius; and shapes are created by smoothly connecting a string of points. Translation between architectures is a straightforward process. A circle can always be recreated if its center point and radius are known. Minor differences in storage methods can be converted using simple geometry. For example, IBM's GFIS software stores an arc as three points (start, center, end) while Intergraph's IGDS software uses a starting point, sweep angle, and radius. Converting between these two systems is possible using the mathematical equations for a circle.

Most systems provide a method for storing redundant graphic representations. These representations are usually stored in a separate symbol or cell library, and identified using a name or number. The graphic file contains only a reference to the appropriate entry in the library, and the representations are retrieved for display when required. Transferring graphics from one system to another can be done by associating the library entry in the old system with the corresponding name or number in the new system. A one-for-one exchange can be made between the two conventions. This same approach can be used to make global changes to the representations.

Non-graphic data elements are also stored in a relatively consistent manner across different system architectures. Minor variations can occur in the technical structure in which the data is stored (ie: alphanumeric vs numeric) but these problems relate to the functional use of the data and not the conversion process.

Minor differences in the physical storage of the data can be overcome using available utility programs. For example, many programs are available to convert between ASCII and EBCDIC. Differences in the storage of numeric data can be accommodated using interface software. For example, a gateway between DEC and IBM must accommodate the fact that numeric bytes are flipped between the two architectures. Differences in the level of precision can also be a factor. Sufficient precision should exist in the old system to facilitate conversion to a more precise system in the future.

3. CURRENT SOFTWARE APPROACHES

Current system architectures utilize one of four physical data structures: sequential, hierarchical, network, or relational. In order of complexity: sequential is simplest; hierarchical is more complex; network is next and relational is the most complex. Assuming future systems will utilize one of these structures, planning for future change requires planning for conversion from one physical structure to another.

Conversion from a complex structure to a less complex structure is relatively straightforward. The conversion from relational to hierarchical dictates that the parentage relationship be identified from other relationships. Conversion to a sequential format entails the writing of records into the correct sequence in a flat file. This conversion is the least complex.

Conversion from a simple structure to a complex structure is much more difficult. Translation from a sequential environment to a hierarchical structure requires building a hierarchy out of sequential data. The hierarchical relationships (parentages) must be established using existing information. Conversion to a relational structure is even more complicated, since all relationships must be determined from available data.

The planning implication for these conversions is significant. In order to prepare for a transfer to an architecture with more complex data relationships, the relationships must be established in the current data.

BEST COPY AVAILABLE

The effort required to create the relationships later may be exorbitant compared to the cost to include them during the current conversion process. It may not be feasible to establish pointers after the data has been loaded, and the files may need to be re-converted.

Current GIS software packages utilize a variety of approaches to tie together graphic and data (non-graphic) elements. In some systems the records are integrated into the same data base structure, and in other systems the records are in separate files which are totally independent of one another. These different approaches represent many challenges when converting from one system to another. Unrelated records may need to be brought together.

Non-graphic records can usually be linked via some common field or relationship. The relationships of these records are often determined by the functional needs of the organization. Most organizations have existing application programs which require the data to be in a certain structure. These programs may dictate the relationships between data elements, and these relationships can be used to restructure the records.

This restructuring process can be difficult when dealing with graphic representations since the ability to link together vector elements can be limited. The graphic relationships may be critical, however, if the software utilizes the graphics to perform certain functions. For example, some packages perform traces through the graphic elements. All connected elements become part of the network. The relationship may be nothing more than geographic proximity, but if the elements are not connected in the old system, then this trace will not be possible in the new system.

GIS software packages may require specific data relationships. The ability to translate from one package to another depends upon the flexibility of the data elements. The primary concern relates to the ability to tie together independent records in separate systems.

Once again the key factor is the existence of a "hook" or unique element which can be used to identify related records.

4. DATA STRUCTURES

Structure within a data base pertains to the physical pointers which relate one data record to another. The logical relationships that these pointers convey are determined during the data base design phase of system development. In most GIS environments the current software dictates the physical structure, but the user determines the logical structure.

Software developments have shown a trend toward increased functionality of the data elements. Increased functionality of the software usually results from increased complexity in the physical structure of the data base. A relational data base can provide greater flexibility than a sequential file.

Users have also discovered that increased functionality can be obtained by increasing the complexity of the logical data structure. It is possible to establish more logical relationships than the physical structure of the data base will permit. Use of color, for example, will distinguish different lines which occupy identical status in the data base structure.

The major concern in changing architectures pertains to the situation where the logical design is not as complex as the physical structure of the new data base. For example, if a hierarchical file is transferred to a relational data base with only the parentage relationships identified, then no increase in functionality will occur. The result will be a hierarchical file operating in a relational environment. The superior functionality of the data base will not be utilized.

In these instances the only way to utilize the increased power of the data base is to add the missing relationships. This entails a re-conversion effort which could be substantial. The effort required to go back and add relationships after the fact is much greater than the effort required to add them the first time around.

The following section reviews the case studies of two companies which were involved in architecture changes. The companies had drastically different levels of complexity in their data base designs. This difference affected the success of the translation efforts, and the corresponding results of their conversion process.

5. CASE STUDIES

The companies described in the following two case studies have almost identical backgrounds. Both companies are electric utilities operating in the United States. Both companies implemented a GIS on one system and then changed to a different architecture after a period of years. The architectures involved were the same in both cases, but the approaches were different due to the individual objectives of each company and the nature of the translation effort.

5.1 Case One

Company One had an existing graphics system which had been in production for a number of years. The graphic representations of the service area were substantially converted. There was minimal attachment of these graphic representations to an intelligent data base infrastructure. The company completed a feasibility study which recommended the purchase of a new GIS system on a different system architecture. As the result of this purchase, the company needed to transfer existing graphic and data files to the new system.

The objective of the transfer effort was to salvage as much existing data as possible and reduce the amount of re-conversion effort which would be required. The functionality of the converted data was to be enhanced as much as possible.

The new architecture was based upon the graphic and attribute records being combined in the same data base. The new system also depended upon the graphic elements to be connected at the same geographic point for tracing purposes. Both of these situations presented problems for the conversion. The existing graphic and data records were basically unrelated, and the graphic representations were not always geographically connected.

In order to establish as much functionality in the new system as possible, an artificial intelligence type of conversion approach was used. Decision rules were established to enhance the non-graphic data, establish connectivity, and build data relationships. Rules were also used to identify the entity types and attach non-graphic attributes to the records as they were brought across. Identifiable graphic elements were connected if they were within a specified proximity to one another. Graphic elements were corrected automatically as much as possible using graphic programming techniques.

Elements which could not be identified or did not fit any of the decision rules were flagged for manual cleanup. These and other remaining errors were resolved at a workstation. Most of the errors were the result of inaccurate data conversion (digitizing) or missing and incomplete records. These errors severely affected the functionality of the new architecture.

An analysis of the results of the conversion showed that the graphic elements were 100% translated but relationships were missing. The non-graphic data was marginally converted, but the accuracy was suspect. The new system was functional, but not able to take full advantage of advanced features. A substantial re-conversion effort was required to gain full functionality in the new architecture.

5.2 Case Two

Company Two had an existing graphics system which was utilized for various mapping and engineering functions. This system fulfilled the requirements of several departments and contained information which was valuable to other users. Due to additional demands for geographically-based information a pilot project was initiated. The pilot system resided on a different architecture and supported a variety of functions outside the scope of the current GIS. Due to the different requirements of the two systems a decision was made to operate both systems concurrently. Since the two systems utilized the same geographic area, an effort was made to minimize the conversion effort required for digitizing the land base. A project was initiated to interface the two systems and transfer data between the different architectures. The existing system would serve as a feeder of geographic data to the pilot system.

The two architectures utilized different approaches regarding graphic and non-graphic data structures. In the existing system the graphic and attribute elements were separated. In the pilot system these elements were combined in the same data structure. The pilot system also had a much higher degree of complexity in its physical data base design. In order to utilize the existing system to provide data to the pilot system, the logical data structure of the existing system had to be changed.

The first step was to define the data elements and relationships in the pilot system. This structure was then compared to the structure of the existing system and missing or incomplete elements were identified. The logical design of the existing system was then modified to include these new relationships. The result was a complex logical design residing within a simple physical structure. The conversion specifications for the existing system were then modified to incorporate the changes identified above, and the existing conversion effort continued.

Data converted using the new specifications was translated into the pilot system architecture and tested for functionality. The results of the translation were a success. The pilot system was fully functional and provided features which were not available in the current GIS.

The final configuration presented some problems, however, due mainly to the concurrent operation of two geographic systems. While only one conversion effort was necessary, the expense was considerably higher than with the original specifications. This represented a significant increase in conversion costs for the existing system. Unnecessary data was also being entered and maintained on the existing system. Duplicate information was maintained on both systems and presented the opportunity for inconsistency in the landbase.

While this final configuration may not prove to be effective for this company, the translation effort was successful and demonstrated that increased functionality can be obtained during a conversion to a different architecture. All graphics and attribute information was completely transferred from the existing system to the pilot system and the pilot GIS was completely functional. No additional effort was required for conversion on the pilot system.

6. CONCLUSION

The goal of this paper was to identify factors associated with a change in system architecture which may affect the future of a Geographic Information System. Data elements and their relationships in current systems can be transferred across different system architectures in a straightforward manner. The different approaches used by current GIS software products vary considerably and present challenges in translation. Records which are not related in one system must be integrated in another system. The major issue regarding translation between systems concerns the conversion from a simple data structure to a more complex set of relationships. The new relationships must be manually defined if they are not present in the current system. This process can be time consuming and costly. A careful review of these issues prior to conversion can provide insurance that the data will have value beyond its current application, and that future system enhancements will be possible without the need to enhance the data.

The two case studies presented were examples of the difference that planning for future changes can have on the total cost of a GIS. An initial low-cost strategy for data conversion may result in a limited future for the converted data. A comparison of the expected life of current systems to the expected life of the geographic data reveals that the data will outlive many generations of software and hardware. There is a possibility that a future software upgrade may entail a change in system architecture, and that the new architecture will require additional functionality from the geographic data.

BLANK PAGE

EVOLUTIONS ON GRAPHIC WORKSTATIONS AND THEIR RELATIONSHIP TO MICRO, MINI, AND MAINFRAME SYSTEMS

Dan Thomas

PlanGraphics, Inc.

Silver Spring, Maryland

ABSTRACT

Over the past two years, there have been dramatic changes in the technology and cost of graphic workstations used with automated mapping and geographic information systems. Decreases in processor cost/performance ratios, trends toward open system architecture, and improvements in system networking capabilities have given rise to supermicro or minicomputer based workstations. Low cost workstations based on microcomputers and "dumb" graphic terminals are also increasing in popularity.

This paper will review recent technological advances and cost/performance changes in graphic workstations and evaluate the impact of these changes on users of automated mapping and geographic information systems.

1. INTRODUCTION

In the continuum of advances in computer technology, and with the tremendous amount of information being generated on the subject, it is easy to overlook major shifts that occur in the industry. These trends may be subtle at first, but eventually provoke a minor revolution, altering previous concepts of a process or way of doing business. Typically, these computer industry revolutions result in increased capabilities and functionality at lower cost. The development of the "intelligent" graphic workstation is one such significant shift.

This document will focus on the development of the intelligent graphic workstation and its effect on the Automated Mapping Geographic Information System (AM/GIS) industry. Specifically, this paper examines the following points:

- The differences between the traditional graphic workstation and the new intelligent graphic workstation

- How AM/GIS software is evolving to take advantage of the intelligent workstation
- How the introduction of the intelligent graphic workstation affects the future of the "dumb" workstation
- The benefits the intelligent workstations bring to the AM/GIS processing environment
- Developments that can be expected to occur in the future and current trends that should continue.

Table 1 lists some of the late-model intelligent graphic workstations available on the market today and the systems that support them. The table identifies each workstation's respective CPU engine and major AM/GIS systems that support them. The table is not comprehensive but is intended to illustrate the breadth of the intelligent workstation market.

TABLE 1
THE NEW GENERATION OF GRAPHIC WORKSTATIONS

Workstation	**CPU**	**Processor Word**	**Supporting Systems**
Apollo Domain	Motorola 68020	32-bit	Auto
DEC GPX	VAX	32-bit	ESRI, GeoV, MDC, Synr
HP 9000	Motorola 68020	32-bit	Delta
Intergraph InterPro 32	Motorola 68020	32-bit	Int
IBM AT	Intel 80286	16-bit	ESRI, GeoB, Synr
Tektronix 4125	Intel 80286	16-bit	ESRI
Sun	Motorola 68020	32-bit	Wild

System Abbreviations

Auto = Autotrol
ESRI = ESRI
Delta = Deltasystems
GeoB = GeoBased Systems
GeoV = GeoVision
Int = Intergraph
MDC = McDonnell Douglas
Synr = Synercom
Wild = Wild-Heerbrugg

2. THE AM/GIS WORKSTATION

What exactly is an intelligent graphic workstation? In order to understand the term, it is first necessary to provide a general definition of the traditional graphic workstation. For purposes of this discussion, a traditional graphic workstation shall be defined as a digital data entry and manipulation device consisting of at least the following components:

- A high-resolution graphics monitor
- A large-format digitizing tablet and cursor
- A keyboard to communicate with the host processor.

The earliest AM/GIS graphic workstations had each of these components, though their capabilities are considered primitive by today's standards. These pioneering devices had monochrome vector screens, minimal resident memory, and no local processing capabilities. They functioned as peripheral devices to the central processor, usually a minicomputer, which was responsible for performing all of the image and data processing tasks. Most AM/GIS vendors designed and manufactured their own proprietary workstation. These stations were quite expensive to purchase and maintain.

Over time, proprietary graphic workstations were improved, adding features such as resident dynamic memory and color raster displays. More recently, proprietary workstations have been enhanced to perform limited local processing, reducing some of the burden on the central computer.

2.1 Intelligent Workstation Development

Within the past two years, a highly flexible, graphic-oriented genre of workstations has emerged on the market. These workstations have their own central processors, high-resolution monitors, and may be configured with large amounts of memory and mass storage. These new workstations are capable of functioning as stand-alone AM/GIS devices, and possess the processing power that was available only from a super-minicomputer not more than three years ago. For the purposes of this paper, these devices have been labeled "intelligent" graphic workstations.

The development of the intelligent workstation is an excellent illustration of the convergence of microprocessors and minicomputer processors. Increased capability of microprocessors and price reductions of minicomputers have made it increasingly difficult to draw a distinction between the two. Currently, the differentiation is usually based upon the processor's lineage. Today's most advanced CPU's that evolved from the personal computer industry are generally considered microprocessors or super-microprocessors. Computers that are based on processors that were first used in larger minicomputers are considered minicomputers.

Intelligent workstations designed around the new generation of super-microprocessors are characterized by 32-bit CPUs such as Motorola's 68020 or Intel's 80386. On the minicomputer end, IBM and Digital Equipment Corp., among others, have launched entries into the intelligent workstation market.

AM/GIS software vendors have recognized the futility of competing with hardware giants such as DEC or IBM, and are beginning to abandon proprietary hardware configurations. These vendors are beginning to exploit the powerful new capabilities and expanded local resources offered by intelligent workstations.

3. AM/GIS SOFTWARE EVOLUTION

The introduction of intelligent wokrstations has paved the way for distributed processing in the AM/GIS application environment. Previously, the centralized processing model was the only choice available to the AM/GIS user. It should be recognized that both the centralized and distributed configuration alternatives are somewhat conceptual in nature, and therefore can be absolute in their definition. In practice, it is improbable that the extremes of a pure central processing or totally distributed processing configuration exist.

3.1 The Central Processing Model

AM/GIS software originated within, and has traditionally revolved around, the central processing model. Within the central processing model, the central host computer is the workhorse that performs all graphic processing tasks. The graphic and attribute databases reside on disk drives directly attached to the single central processor. Operators load and maintain the database using traditional workstations (described above). Each wait their turn for processing resource and database access via the central processor.

This single processor model is neat and orderly from a programmer or database manager's point of view, because access to the database is easy to manage and control. However, from a performance perspective this configuration is inefficient. Depending on the particular computer involved, the system may quickly become overwhelmed as processing load increases through the addition of workstations or resource usage. For this reason it has been important to initially acquire a computer with sufficient capacity to drive the number of workstations that would eventually reside on the system.

If a project outgrows the processing capabilities of the central host computer, it is very expensive to migrate to a larger CPU. The expense of migrating to an upgraded processor is not limited to cost of the new hardware and installation. A significant portion of the outlay is allocated toward purchase of software license upgrades for existing software to run on the new CPU.

3.2 The Distributed Processing Model

The new intelligent workstations present an AM/GIS software producer with the opportunity to exploit the distributed processing environment. Effectively each operator can have the full functionality of the minicomputer at his/her desk at significantly lower cost than the traditional proprietary workstation.

Integrating the intelligent graphic workstation is a significant task for vendors whose systems use traditional workstations. Aside from the obvious task of developing device drivers to support new equipment, software producers have had to redesign some fundamental system operations. The distributed processing environment raises new problems for the vendor to solve. Some major issues that must be resolved include:

- What portion of the application software should be local to the workstation?
- How will the system manage data that may be physically located on different computers in a real-time manner?
- How will the system avoid unnecessary data duplication and potential data corruption across multiple processors in the network?

Fortunately, the leaps in processing technology have been accompanied by gains in sophisticated networking software. Many of the overhead tasks required for a network this closely integrated are performed for the AM/GIS vendor by the new generation of communications software.

4. The Demise of the "Dumb" Terminal?

Wide availability of the new inexpensive intelligent workstations would seem to have certainly signaled the end of the traditional, or "dumb", graphics terminal. However, this assumption is incorrect. Though "inexpensive" may be used to describe the intelligent workstation, it is a relative term. These powerful processors cost in the range of $15,000 to $50,000; certainly reasonable when compared to the cost range - $25,000 to $75,000 - of their proprietary predecessors.

Since not all AM/GIS system users require the power of these intelligent workstations, there is quite clearly a need for the less expensive, less functional station for simple database query and access tasks. "Dumb" terminals fit this niche quite nicely. They are becoming less expensive ($7,000 to $15,000) and also are gaining higher graphic resolution. These terminals range from single purpose graphic terminals such as the Tektronix 4107 or 4109, to multiple function personal computers like the IBM AT configured with a high-resolution graphics card and a monitor. It is clear that the "dumb" terminal will continue to see use in the AM/GIS environment.

5. Benefits to the AM/GIS Community

What do the advancements in graphic workstations and AM/GIS software mean to the end user? Several benefits to the user community have been realized as a direct result of the emergence of the intelligent graphic workstation.

- Easier entry into the AM/GIS market - the low cost and stand-alone power of the new workstations have encouraged several AM/GIS software vendors to lower licensing fees on their products. The lower fees make advanced AM/GIS systems available to groups that previously could not afford them.
- Easier system migration path - the distributed processing philosophy behind the new intelligent stations allows for system additions and expansions in smaller increments and much lower costs than previously available. When it becomes necessary to increase system power, another workstation may be added to the network to reduce processing bottlenecks. This incremental approach allows the customer to add hardware capacity where it is lacking.
- Management of data ownership - the distributed processing model encouraged by the new stations offers unique opportunities for decentralizing data entry and maintenance tasks.
- Open architecture - as software producers continue to broaden the menu of hardware supported by their systems, options for the customer will increase. Support of third-party equipment encourages standardization and protects the investment of the customer. The user community will also benefit from the efficiency of hardware

BEST COPY AVAILABLE

manufacturers performing hardware research and development, rather than a parallel yet less productive effort by software producers.

- No need for controlled environment - as computers have become more compact they have become more rugged, with reduced heat output and more tolerance to being moved about. Minicomputers, like mainframes, have traditionally required a filtered, air conditioned environment. The new stations with processing capacities rivaling the older minicomputers now quietly coexist in the office environment under a desk.

6. Looking to the Future

In summary, trends related to the introduction of intelligent graphic workstations that can be expected to continue are:

- Hardware performance/cost ratio will continue to increase
- The distinction between microcomputers, minicomputers, and mainframes will continue to erode
- The trend away from proprietary hardware configurations toward third party workstations produced by manufacturers such as Apollo, DEC, IBM, Sun, Hewlett Packard, Tektronix, and others is expected to continue
- Intel's 80386 processor, IBM's choice for the most advanced models of its PS/2 product line, will bring true 32-bit processing power to the mass market in the form of personal computers. These systems will provide, to the general public, the horsepower necessary for sophisticated graphic processing.

BLANK PAGE

Space Efficient Hierarchical Structures : Relatively Addressed Compact Quadtrees for GIS

Peng Gao
Terence Smith
The University of California, Santa Barbara, CA 93106.

Abstract

Although they possess good hierarchical structure, the traditional pointer-based quadtree data structures are considered inappropriate for applications in GIS, when compared with linear quadtree data structures. This has been due to their space inefficiency and the fact that they are hard to manage on secondary storage. In this paper, improved pointer-based quadtrees, called Relatively Addressed Compact Quadtrees, are presented which provide a hierarchical data structure and simultaneously solve the problems associated with traditional pointer-based quadtrees. The new data structure uses a relative addressing scheme instead of an absolute one to reduce the storage requirements of the quadtree pointers and employs a special coding method to fully utilize the space allocated for unnecessary pointers. Analysis and experiments show that the storage requirements of the improved quadtree is comparable with the linear quadtree for binary images. For multi-colored images, average storage requirements are less than 2 bytes per node, with each node storing minimum, maximum and mean values. A memory management scheme is also proposed for managing the pointer-based quadtree in secondary storage. It is shown that this memory management scheme is quite efficient when applied to the quadtree based GIS.

1. Introduction

The quadtree of a 2^n by 2^n image is defined by recursively subdividing the image into quadrants, subquadrants, and so on, until blocks (possibly single pixels) with uniform value are reached. The subdivision is represented by a tree of degree 4, called the Regular quadtree, also referred as the Pointer-based quadtree. The root node corresponds to the entire image, the four sons of a node are its quadrants, and terminal nodes represent blocks with a single value. Each node of the tree is stored as a record with six fields. Four fields contain pointers to the four sons of the node; one field contains a pointer to the father of the node; and the last field stores color information.

The Regular quadtree has several advantages over other types of quadtrees. The pointer system of the Regular quadtree provides an efficient way to move within the quadtree so that certain operations, such as connected component labeling [5, 6, 7], union/intersection [2] as well as spatial search [8], can be efficiently performed. Hierarchical information can be stored at each level of a multi-colored quadtree allowing the dynamic computation of the color of a node with respect to some spatial object whose location is being sought. Spatial search on multi-colored images can then be implemented in an efficient top-down fashion.

Though it has a pointer system and a good hierarchical structure, the traditional Regular quadtree is considered inappropriate for applications in Geographic Information Systems when compared to linear quadtree data structures. This is mainly because of its relative space inefficiency and the fact that Regular quadtrees are hard to manage on secondary storage. The Regular quadtree requires five pointer fields for each node. The size of these pointers must be large enough, normally 4 bytes, to address any location within a data base. On the other hand, only one field is used to store the description of the nodes. In most implementations of Regular quadtrees, each node takes about 21 bytes of memory. Only one of these 21 bytes is used to store color information. The memory needed to store pointers is thus far greater than the memory required to store node color. This problem has been recognized by many GIS researchers and much effort has applied in attempts to overcome the deficiency. Among the data structures that have resulted from this research are two important improved pointer-based representations, called One-to-four quadtrees and Autumnal quadtrees, respectively. Both of these quadtrees successfully reduce storage requirements.

The One-to-four quadtree was first proposed by I. P. Stewart in 1986. The One-to-four quadtree is so named because one single pointer is used to determine the location of the records for all the sons of four quadrant descriptors [9]. The basic building block for the One-to-four is a record representing four sibling quadrants or nodes. This record consists of five fields. Four of the fields store descriptors, for each of the sibling quadrants. The fifth field is a pointer to a block of four child records that store information on all the sons of the four quadrants in the record. The One-to-four thus substantially reduces the storage requirement of pointers since each pointer in the One-to-four replaces sixteen pointers in the Regular quadtrees. Such a saving of memory, however, is obtained by sacrificing time efficiency. Significant amounts of computation are involved in order to locate the record storing information on the sons of a specific quadrant.

The Autumnal quadtree (AQT) is another improved pointer-based quadtree structure that not only retains the advantages of the Regular quadtree but also achieves storage savings. The basic idea of Autumnal quadtrees is the elimination of storage for leaf nodes in the Regular quadtree. This is done by encoding the values of leaf nodes into the unused pointer space of gray nodes. By storing only gray nodes, Autumnal quadtrees achieve a 75% storage savings over Regular quadtrees without losing efficiency [4]. Although coding leaf nodes makes use of the space wasted on pointers, replacing pointer information by one-bit information (BLACK or WHITE) does not fully utilize pointer space. There is thus the potential for more storage savings when using Autumnal quadtrees. As the image size becomes large, pointers still takes the major portion of the storage for Autumnal quadtrees as they retain five pointers.

GIS are intended to handle very large, multilayered, heterogeneous databases of spatially indexed data. This implies that the amount of data in a GIS is so large that it is impossible to store the whole database in main memory. File management must be employed so that at any instant of time, only a portion of the data base is resident in main memory and the rest of the data base is stored in external storage as a set of files. Whenever requested data are not in main memory, I/O functions are required to fetch the requested portion of the data base. Compared with linear quadtrees, where the B tree or B^+ tree file structures are well developed [1, 3], pointer-based quadtrees have not found a suitable file organization to store the complete quadtree in external storage. The most commonly used method is based on a traversal of the pointer-based quadtree, converting it into a linear quadtree which is then stored in place of the original pointer-based quadtree, in external storage. Each time the quadtree or part of the quadtree is requested and fetched into main memory, the pointer system must be reconstructed from the corresponding linear quadtree. The format conversion and quadtree reconstruction costs are not trivial when the data base is very large.

In this paper, the two major problems associated with pointer-based quadtrees are further examined and better solutions are proposed. A new space efficient pointer-based quadtree for binary images is presented in Section 2. This quadtree structure, called the Compact quadtree (CQT), achieves more storage savings than the One-To-Four and Autumnal quadtrees. An extension of the binary CQT to multicolored images is given in Section 3. A Relative Addressing scheme is introduced in Section 4. By dividing the entire image into a set of subimages and using relative addresses within each subimage instead of absolute ones, this scheme not only reduces the size of quadtree pointers, thereby reducing storage requirements, but also makes it possible to store the complete pointer-based quadtree, not the linear version, in external storage. This eliminates quadtree reconstruction costs. A spatially indexed file organization is presented in Section 5. This file organization is relatively simple but very efficient for most common GIS operations. Finally, conclusions are proposed in Section 6.

2. The Binary Compact quadtree

The CQT is essentially derived from the basic ideas of both the One-to-four and the Autumnal quadtrees. It employs a modification of the basic One-to-four idea to organize pointers and uses the encoding technique from the Autumnal quadtree to handle leaf nodes. In this way, the CQT results in even more storage savings.

2.1. Basic Structure of the Compact quadtree

The first step in deriving the CQT is to modify the One-to-four quadtree into a One-to-one quadtree by adding three more pointers to a One-to-four record. Recall that the basic building block of the One-to-four is a record representing four quadrants. This record consists of five fields, only one of which is a pointer. This fifth field is used to point to the set of four records that store information on all the sons of the four quadrants. The One-to-one quadtree is similar to the One-to-four except that three more pointers are added to a basic building block. Each quadrant in a record thus has its own pointer that points directly to the record containing its four sons. The three new pointers cause two changes in the One-to-four quadtree. First, the computation required to locate the record storing information on the four sons of a specific quadrant is eliminated, since each quadrant contains a separate pointer to its sons. There is thus a one to one correspondence between records and pointers. Second, the storage requirement for pointers is increased because of the three extra pointers.

The second step of the derivation is to apply the encoding technique of the Autumnal quadtree to the One-to-one quadtree. The basic idea of the Autumnal quadtree is to codify leaf nodes into pointers in their father's record and pointers can be used to represent record positions or quadtree node colors.

Applying this idea to each quadrant in the One-to-one record, it is clear that value fields in each One-to-one record are no longer necessary and the relevant information in each value field can be coded into the pointer field associated with the quadrant. This quadrant coding of the One-to-one results in the first part of the storage savings of the CQT. It is easy to show that the storage requirement of the CQT resulting from this quadrant coding is exactly the same as that of the original Autumnal quadtrees.

In addition to the quadrant coding, there is another level of coding applied to each record that represents four quadrants in the One-to-one quadtree. The basic building block of the One-to-one quadtree is a record instead of a node and each such record has a pointer pointing to it in the record containing its father. Analogous to the leaf nodes in the Regular quadtree, we define a Leaf Record in the One-to-one quadtree as a record of four quadrants in which each quadrant has a single value (BLACK or WHITE). Hence, if a record is a leaf record, the record, as a whole can also be coded into the pointer field associated with its father. The leaf record coding however, is slightly different from the leaf node coding because each pointer in a record is used to represent all four sons of a quadrant instead of only one as in the Autumnal quadtree. The leaf record coding, therefore, can be performed only with the leaf records.

The leaf record coding undoubtedly can involve greater storage capacity than the original Autumnal quadtree. However, because the coding can only be used on leaf records and the number of leaf records varies with different data distributions, the actual storage savings of the leaf record coding scheme depends on the actual data stored in the data base. Experiments indicate that, on average, leaf record coding can achieve an additional 45% saving in storage.

After the two steps of modification described above, the basic record structure of the CQT is finally constructed as a record with only four pointers to each of four child records. In the resulting record each of the four pointers has the following meaning:

(a) a positive pointer represents a link to a child record;

(b) a null pointer implies the existence of a WHITE leaf sons node;

(c) a negative pointer with value -1 implies the existence of a BLACK leaf sons node;

(d) a negative pointer with value other than -1 indicates the existence of a coded leaf record.

How a leaf record is coded into a pointer is left unspecified. The detailed coding configuration can be arbitrarily chosen as long as space exists for it.

2.2. Assessment of Storage Requirements

Let T be the total number of nodes and G the number of GRAY nodes in a Regular quadtree. In a quadtree the following relation holds: $T = 4 * G + 1$. The number of elements N (nodes, records or locational codes) required in the different coding schemes will be used as the measure of storage requirement and is represented as a function of T in the following storage cost calculation:

(a) the number of records in the One-to-four quadtree is

$$N = 0.25 * T$$

because each record includes four Regular quadtree nodes;

(b) the number of nodes in the Autumnal quadtree is

$$N = 0.25 * (T - 1)$$

since the AQT only stores GRAY nodes;

(c) the number of records in the CQT depends on the actual data distribution. In the best case where all leaf nodes (BLACK and WHITE) are contained in leaf records, the number of records is

$$N = 0.25 * (0.25 * T - 1).$$

In the worst case where no record contains four leaf nodes, the number of records is

$$N = 0.25 * T.$$

Table 1 summarizes the amount of storage required by several different types of quadtrees. For purpose of comparison, the Regular quadtree and Linear quadtree are also included in the table. The total number of the nodes of the RQT is taken as 1. The two entries for the LQT and CQT are referred to the best and the worst cases, respectively. The average storage requirements were obtained from an experiment on 10 different data layers.

The following assumptions have been made for calculation of the storage requirement of different types of quadtrees:

(a) each RQT node takes 9 bytes, 8 bytes for four pointers and 1 byte for color information;

(b) each LQT locational code needs 4 bytes;

(c) each One-to-four record needs 8 bytes, 4 bytes for color information, 3 bytes for pointer and offset as well as 1 byte for extra information;

(d) each AQT node takes 8 bytes for its four pointers;

(e) each CQT record also needs 8 bytes.

It should be emphasized that the average storage requirement of the CQT is an experimental result from 10 data layers of Land Use and Geology of a spatial database for the Black Hills, S. Dakota. The number of records in each data layer varied from 52 to 7457.

Table 1. The amount of storage required by different quadtrees

QT Types	Elements N	Storage (Bytes)	Average Storage (Bytes)
RQT	1	9	-
LQT	0.18-0.54	0.72-2.16	1.4
One-To-Four	0.25	2	-
AQT	0.25	2	-
CQT	0.0625-0.25	0.5-2.0	1.1

3. The Multi-colored Compact Quadtree

The major advantage of using multi-colored quadtrees is to provide hierarchical data structure. Storage of hierarchical information at each level allows the dynamic computation of the color of a node. Past experience in developing a Knowledge Based Geographic Information System (KBGIS-II) have shown that a good hierarchical data structure is essential for fast top-down search in the quadtree-based GIS [8].

The extension of the binary CQT to a multi-colored one follows immediately upon adding a quadrant descriptor for each quadrant in a record. In order to provide sufficient information for fast top-down spatial search, the quadrant descriptor in our implementation for continuous data contains maximum, minimum and mean values, taking a total of three bytes. Since the multi-colored image needs more space to represent this color information, which is normally stored in one byte for a binary image, the quadrant and leaf record coding used in the binary CQT need to be modified for multi-colored images.

For a leaf node the maximum, minimum and mean value collapse to a single value. There is thus no problem with quadrant coding because a pointer is large enough to contain the color information for one leaf node, which is a single value. The problem occurs when the record coding method is applied to multi-colored images because the pointer space (two bytes) is too small to hold all the color information (four bytes) of a leaf record. Therefore, a new record coding method is needed for multi-colored image. At first, it seems that a larger pointer (at least four bytes) is needed in order to codify a leaf record into its father's record. However, it is actually not necessary because some information is repeated in the leaf record and in the record that contains its parent quadrant.

Consider a quadrant with four sons, all of which are leaves with different values. The parent record which stores the quadrant descriptor stores the minimum, maximum and mean for this quadrant. Now consider the leaf record which stores the four sons of the quadrant. The information in this leaf record is to be codified into the pointer space associated with the quadrant in the parent record. The information to be codified consists of four values, one for each son in the leaf record. Two of these values are however already stored as the minimum and maximum values for the quadrant, in the quadrant descriptor in the parent record. There is, therefore, no need to code these two values again when pulling the leaf record information into the pointer space in the parent record. The remaining two node values, are easily stored in the space allocated for the pointer and the mean value.

Under this new record coding method, the pointers in each record have the following meaning:

(a) a positive pointer represents the link to its son's record;

(b) a pointer with value from -255 to 0 represents a leaf node with color information;

(c) a pointer with value smaller than -255 indicates that a leaf record is coded into the pointer and quadrant descriptor. Two values are stored in the spaces for the maximum and the minimum values and two other values are coded into the spaces for the mean value and the lower half of the pointer. The upper half of the pointer is used to determine the allocation of the four coded values among the sons of the quadrant.

Table 2. The Amount of Storage Required by Multicolored CQT

Layer Name	Elements N	Storage (Bytes)
Aspect	0.0625	1.25
Slope	0.0625	1.25
Elevation	0.07	1.40
Average	0.065	1.30

The total number of records in multi-colored CQT is exactly the same as in a binary quadtree. The number of bytes per record in the case of multi-colored quadtrees is 20 according to the particular structure implemented in the KBGIS-II. The storage requirement is 1.25 and 5.00 bytes in the best and the worst cases, respectively using the same calculation as in the last section. The experiments on three data layers, Elevation, Slope and Aspect, have shown that the actual average storage requirements are 1.30 bytes per quadtree node (see Table 2).

4. A Relative Addressing Scheme for Pointer-Based quadtrees

4.1. The Definition Of Relatively Addressed Compact quadtrees

The Relatively Addressed Compact quadtree (RCQT) of a 2^n by 2^m image consists of a root image of 2^{n-k} by 2^{m-k} and a set of regular CQT defined on a set of subimages of 2^k by 2^k, where n, m and k are integers and $n \geq k, m \geq k$. The set of regular CQT is obtained by subdividing the image into a set of subimages of 2^k by 2^k and then constructing a separate CQT for each subimage. The pointers in the quadtrees are all relative addresses based on the upper-left corner of a subimage, not absolute addresses within the entire image. Each member in the set of regular CQT is referred to as a subtree. A cell at position (x, y) in the root image of 2^{n-k} by 2^{m-k} represents the root of a CQT whose upper-left corner is located at $(x2^k, y2^k)$ with respect to the entire image.

According to the definition, two parts of the address, the subtree address and the relative address within the subtree, are needed to locate any node in a relatively addressed quadtree. The subtree address can be found immediately in the root image because the root image is a two dimensional array and indexed by location. By traversing the specified subtree, any node can be located.

4.2. The Advantages Of Relatively Addressed Compact quadtrees

There are at least two major advantages of the relative addressing scheme over the absolute one. These advantages include space efficiency and ease of organization in external storage. It should be noted that the advantages described in the following are not only true for CQTs but also for other types of quadtrees, if the relative addressing scheme is used. The relative addressing scheme is introduced using the CQT.

The quadtree structures described in the last two sections have serious limitations in terms of their capability of representing large images in very large scale GIS although they are more compact than linear quadtrees. According to the assumptions made in Section 2, the pointer size in the pointer-based quadtrees takes two bytes (16 bits). This means that all the quadtree structures listed in Table 1, except the linear quadtrees, can only represent images with a size smaller than 256 by 256 because the maximum positive integer representable by a two byte word is 32767 which is smaller than 256 * 256. In order to handle an image with size larger than 256 * 256, the pointer size therefore needs to be enlarged. For a 4096 by 4096 image, for example, the pointer size required is twice as large as the old one, i.e. four bytes in practice. As a result, the total storage requirement for the quadtrees is also increased. In particular, the storage requirement for the CQT is 2.2 bytes instead of 1.1 bytes, which on average is no better than the requirements for linear quadtrees. This problem occurs because pointer size is a function of image size. For an image of 2^n by 2^m, the relationship between the image size and the pointer size l is $l \geq n+m$.

However, in the relatively addressed quadtrees, each subtree is defined on a small subimage. The size of the subimage is selected independent of the image size and does not change as the image size changes. The image size information is reflected only in the root image of a relatively addressed quadtree. Therefore, by selecting a proper subimage size k, the analysis in Section 2 still holds. This means, in this particular case, that the relative addressing scheme achieves a 50% saving in space over the absolute one for representing an image of the same size.

It is almost always the case that the savings on storage are obtained at the expense of time efficiency. However, this is not true in the relatively addressed quadtrees because the only difference between the two alternate addressing schemes is that the relatively addressed quadtree uses relative addresses instead of absolute ones. The basic quadtree structure is thus unchanged in the relatively addressed quadtree.

The relatively addressed quadtree can also be viewed as a hybrid of the raster and quadtree structures. The top portion of the relatively addressed QT essentially is a raster image with each pixel representing the root of a subtree. The lower portion of the quadtree is a set of subtrees with each being rooted in the root image. This logical division of a quadtree into a root image and a set of subtrees provides a natural way to store the relatively addressed quadtree in external storage. The root image can service as a file index and provide random access to each individual subtree. Each subtree is stored separately as one I/O unit and pointed to by an entry in the file index.

Because the size of a subtree is relatively small, it is possible to Read/Write a complete subtree from/to external storage as one unit. By using a special dynamic memory management method, the pointers in a subtree are represented as an offset from the beginning of a continuous memory area and embedded in the subtree. When the subtree is written to external storage, in addition to node color information, the embedded pointers are also stored in external storage. Each time a subtree is read from external storage, a complete pointer-based subtree is loaded and therefore no reconstruction of the pointer system is necessary.

5. An External File Organization for Pointer-Based quadtrees

The file organization discussed in this section is specially used for managing relatively addressed quadtrees in external storage. The basic organization consists of two parts, a file index and a set of variable length records. Each entry in the file index corresponding to a cell in the root image includes the root information about a subtree and the position where the subtree is stored. Like a raster image, the file index is organized as a two dimensional array and stored row by row. Random access is available only to each row and each individual entry in the file index is not addressable. A record in the file contains a complete pointer-based quadtree corresponding to a subtree in a relatively addressed quadtree and is stored in a set of continuous external memory blocks. Both node color information and pointers are all stored in the record. The position of the variable length record is indicated by its index entry in the file index. A node is retrieved into main memory by fetching the subtree of which it is a member. The overhead in retrieving an entire subtree to examine a node during any procedure is offset by the expectation that nodes visited by operations are spatially adjacent.

The basic operations on this file organization are assumed to be Read/Write a specified subtree, or record from/to a disk file. The cost of retrieving a particular subtree is 1 disk access in most cases and 2 disk access in the worst case. The file index is assumed large enough so that at any time only a certain part of it is resident in main memory. Which portion of the file index resides in main memory at a given time depends on the access history of the file. If the part of the file index to which the requested subtree belongs is not in main memory, then that part of the file index must first be read into main memory. This process takes one disk access because the file index is randomly accessible by row. If the desired file index is already in, or is read into, main memory, the specified subtree can be retrieved by a single disk access using the record pointer stored in the file index.

The storage utilization of this file organization is affected by several factors including the subimage size k, the data distribution in a subtree as well as the external memory block size. In the current implementation, experimental results have shown that the average storage utilization is about 70%.

6. Conclusions

The CQT, combined with the relative addressing scheme, substantially improves the space efficiency of pointer-based quadtrees. The relatively addressed CQT is so space efficient that it is even better than linear quadtrees, which are widely used in quadtree based GIS. The relative addressing scheme is very important to the improvement of the pointer-based quadtrees. It not only reduces storage requirements, but also makes the pointer-based quadtree easy to manage in external storage. The file organization proposed in this paper is simple but quite efficient in terms of both access cost and storage utilization. The relative addressing also makes it possible to store the complete pointer-based quadtree, including both color information and pointers, in external store and therefore no quadtree reconstruction is needed. The final conclusion to be drawn from the various improvements discussed in this paper is that pointer-based quadtrees have now become a suitable hierarchical data structure for GIS, even on a very large scale.

Reference

1 *D. J. Abel,* A B^+-Tree Structure for Large Quadtrees, Computer Vision, Graphic and Image Processing, Vol.27, PP.19-31, 1984.

2 *F. W. Burton, V. J. Kollias and J. G. Kollias,* A General PASCAL Program for Map Overlay of Quadtrees and Related Problems, The Computer Journal, Vol.30, No.4, 1987.

3 *D. Comer,* The Ubiquitous B-Tree, Computing Survey, Vol.11, No.2, June 1979.

4. *F. Fabbrini and C. Montani,* Autumnal Quadtrees, The Computer Journal, Vol.29, No.5, 1986.

5 *H. Samet,* Connected Component Labeling Using Quadtrees, Journal of the Association for Computing Machinery, Vol.28, No.3, July 1981.

6 *H. Samet,* Neighbor Finding Techniques for Image Represented by Quadtrees, Computer Graphic and Image Processing, Vol.18, PP.37-57, 1982.

7 *H. Samet,* A Top-Down Quadtree Traversal Algorithm, IEEE Transaction On Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No.1, 1985.

8 *T. R. Smith, D. J. Peuquet, S. Menon and P. Agarwal,* KBGIS-II: A Knowledge Based Geographic Information System, International Journal of Geographic Information System, Vol.1, No.2, 1987.

9 *I. P. Stewart,* Quadtrees: Storage and Scan Conversion, The Computer Journal, Vol.29, No.1, 1986.

GIS PROCESSING IN THE NATIONAL PARKS-

THE UNIX ENHANCED PC ENVIRONMENT

Harvey Fleet and Sharon S. Shin

Geographic Information Systems

National Park Service

Introduction

During the past decade GIS technology at the National Park Service has grown slowly and steadily. Very recent developments, however, are leading to a boom in the use of the technology. Aided by active and continuing system development, national parks are finding GIS applications in their daily and long term management requirements.

System development has been difficult: parks have insufficient funds to acquire GIS technology, to construct data bases, to perform GIS activities, and to acquire hardware. Until recently the computer required for GIS processing financially beyond the reach of most parks. The availability of UNIX coprocessors for PC-level machines has changed this situation. This paper describes the approach the National Park Service is taking to this hardware opportunity.

GIS and the National Park Service

The National Park Service administers over seventy-nine million acres of the nations most naturally, culturally, and historically significant land. Our mission is to provide recreational use while preserving these treasures for future generations. Park Service managers are discovering the potential of GIS in meeting these responsibilities. At this writing, twenty-eight parks have created or are constructing data bases. Double that number are planning GIS activities.

Unfortunately, on site GIS activities are fewer in number, for two principal reasons: inadequate commitment of personnel and lack of suitable hardware. Commitment of personnel is not a technological problem and will slowly resolve itself, especially when GIS proves its value to management. About six parks have committed staff to GIS activities, ranging from a few hours per week to two full-time employees per week.

The hardware barrier is falling, however, due to rapid and continuing advances in hardware development. Until recently the only viable options for in-park GIS processing were based on 16-bit technology or the use of time-sharing--linking to a remote mainframe computer. Neither of these options has proved satisfactory to the parks, due to extended processing time and limited mass storage or cost of time sharing and long distance telephone rates. As a result, only a few parks have adopted GIS activities.

TABLE 1

NATIONAL PARKS WITH OPERATIONAL GIS CAPABILITIES

Santa Monica Mountains National Recreation Area (CA)

Yosemite National Park (CA)

Redwoods National Park (CA)

Obed Wild and Scenic River (TN)

Biscayne National Park (FL)

Affordable thirty-two bit technology has changed this condition. Thirty-two bit processors for PC-level machines and workstations has made it possible to place mainframe computing power into the hands of field users throughout the National Park Service. With the potential of over 300 sites we are challenged to find suitable and affordable software that can take advantage of the hardware opportunity. Utilizing public domain software in the UNIX environment allows NPS to meet the challenge. Public domain software frees us from licenses, fees, and proprietary rights and allows us to distribute, modify, and enhance software freely.

UNIX Hardware

At this writing, the NPS is implementing UNIX processing in two ways: through integrated UNIX workstations or through UNIX coprocessor boards that are inserted into existing IBM-compatible PCs. The latter configuration, which we call the UNIX-enhanced PC, will be discussed in the remainder of this paper.

The UNIX Processor Board

The UNIX processor board is a full function UNIX compute engine. It gives the PC full access to the speed and power of the thirty-two bit environment. The PC provides mass storage, power supply, communications, and access to the DOS environment. The PC is effectively transformed into a UNIX work station, while maintaining access to the DOS environment. Of the several boards on the market, we use one based on an NSC 32032 chip set. It utilizes AT&T System V UNIX, has four megabytes of physical memory, sixteen megabytes of virtual memory, and a nominal clock speed of sixteen megahertz. Much more powerful boards are available.

Public domain software and the UNIX board.

The GIS Division of the National Park Service utilizes and supports two public domain software systems: SAGIS and GRASS. Together these software systems provide a complete set of GIS capabilities, including basic image processing (GRASS). Both run under UNIX: SAGIS is written in FORTRAN77, GRASS in C. These software systems have virtually all the GIS processing capabilities that parks will require for the foreseeable future.

Other Requirements

Suitable display and mass storage are important factors in transforming the UNIX-enhanced PC into a workstation. Although display options are numerous, our software places two important restrictions on the requirements: Tektronix compatibility (specifically the 4100 series) and two-hundred fifty-six simultaneously displayable colors (for raster operations dealing with elevation and imagery data). Hardware meeting these requirements is available at reasonable cost.

Mass storage options are numerous. We favor removable, random access media, such as Bernoulli drives, which can provide up to forty megabytes of on-line (but removable) storage per dual drive. Higher volume devices, based on Winchester or optical technology, can also be readily utilized.

TABLE 2.

GENERIC CONFIGURATION FOR A UNIX-ENHANCED GIS WORK STATION

IBM-compatible PC/AT (approximate cost $3000)

UNIX coprocessor board (a.c. $3500)

Sufficient mass storage (e.g., Bernoulli drives (a.c. $2300))

Graphics display device (a.c. $3000)

Color screen copier (a.c. $1300)

Digitizing tablet (a.c. $3000)

Park Activities.

Several parks have acquired or are acquiring UNIX-enhanced PCs. Applications vary widely but typically range over resources management, environmental assessment, and operations management. It is still too early to estimate the cost effectiveness of the technology as a management tool in the NPS. Only time will tell whether the technology is an effective and useful tool for managing park resources.

TABLE 3

NATIONAL PARKS OR RELATED UNITS THAT HAVE ACQUIRED OR ARE ACQUIRING UNIX - ENHANCED PC WORKSTATIONS

George Washington Memorial Parkway (D.C.)

Mount Rainier National Park (WA)

Southeast Archeological Center (FL)

Gulf Islands National Seashore (FL MS)

Glen Canyon National Recreation Area (UT)

Bryce Canyon National Park (UT)

Shenandoah National Park (VA)

References.

Fleet, Harvey, "GIS Technology in the National Park Service." Paper presented to the Ninth Annual Applied Geography Conference, West Point, N.Y., October 23-25, 1986.

_________. "SAGIS: A Full-Function Public Domain GIS for Micro- and Mini Computers." Proceedings of the Geographic Information Systems Workshop. American Society of Photogrammetry and Remote Sensing, Atlanta, GA., April 1-4, 1986.

_________. "GIS Trends and Applications within the National Park Service." Paper presented to the Nineteenth Symposium and Workshops on Surveying and Photogrammetry of the Rocky Mountain Region of the ASPRS and Colorado Section of the ACSM. Denver, CO., January 13-17, 1986.

_________. "Geographic Information Systems and Remote Sensing Activities in the National Park Service." Proceedings of Geographic Information Systems in Government." Bruce K. Optiz, ed., U.S. Army Engineer Topographic Laboratories, Fort Belvoir, VA., December 10-13, 1985.

National Park Service, The National Parks: Index 1985. U.S. Department of the Interior. 1985.

Westervelt, James and Goran, William., "Software and Work station Design Considerations for GRASS." Technical Papers Proceedings of 1987 ASPRS - ACSM Annual Convention. Volume 1 pp 219-226.

Baltimore, MD. March 29 April 3, 1987.

Westervelt, James and Goran, William., Grass Users Manual, Version 2.0. U.S. Army Construction Engineering Research Laboratory, Champaign, IL. 61820.

BLANK PAGE

☆U.S. GOVERNMENT PRINTING OFFICE:1980-220-010

END

DATE FILMED

12-18-91